信创大赛-备份与还原篇

武汉达梦数据库有限公司内部资料

达梦数据库信创大赛资料

目录

目录

1 物理备份还原简介......1
1.1 基本概念......2
1.2.1 表空间与数据文件......2
1.2.2 重做日志......3
1.2.3 归档日志......4
1.2.4 备份......5
1.2.5 还原与恢复......7
1.2 备份方式......9
2 物理备份还原......9
2.1 归档说明......9
2.1.1 本地归档......9
2.1.2 归档切换......10
2.2 备份......10
2.2.1 数据备份......11
2.3 还原与恢复......12
2.3.1 数据还原......12
2.3.2 数据恢复......14
3 物理备份还原实战......16
3.1 准备工作......17
3.1.1 支持与限制......17
3.1.2 归档配置......19
3.2 使用联机执行 SQL 语句进行备份还原......22
3.2.1 数据库备份......22
3.3 使用脱机工具 DMRMAN 进行备份还原......24
3.3.1 DMRMAN 概述......25
3.3.2 启动 DMRMAN......26
3.3.3 数据备份......26
3.3.4 数据库还原和恢复......27
3.3.5 查看操作日志......34
3.4 使用图形化客户端工具进行备份还原......35
3.4.1 使用 MANAGER 工具进行联机备份还原......35
3.4.2 使用 CONSOLE 工具进行脱机备份还原......45
4 逻辑导入导出功能简介......55
5 dexp 逻辑导出......57
5.1 使用 dexp 工具......57
5.2 dexp 参数一览表......57
5.3 dexp 参数详解......60
5.3.1 特殊参数......60
6 dimp 逻辑导入......64
6.1 使用 dimp 工具......64
6.2 dimp 参数一览表......65

6.3 dimp 参数详解......67
6.3.1 特殊参数......67
7 dexp 和 dimp 应用实战......69
7.1 一个完整示例......69
1．环境准备......69
2. dexp 导出......70
3．dimp 导入......70
4．查看 LOG 日志......70

达梦数据库信创大赛资料

1 物理备份还原简介

数据库备份是 DBA 日常最重要的工作内容。备份的主要目的是数据容灾，保证数据的安全性，在数据库发生故障时，通过还原备份集，将数据恢复到可用状态。本章主要介绍 DM 物理备份与还原的概述和基本概念，为理解归档配置、备份还原的原理与工具使用奠定基础。

DM 数据库中的数据存储在数据库的物理数据文件中，数据文件按照页、簇和段的方式进行管理，数据页是最小的数据存储单元。任何一个对 DM 数据库的操作，归根结底都是对某个数据文件页的读写操作。

因此，DM 备份的本质就是从数据库文件中拷贝有效的数据页保存到备份集中，这里的有效数据页包括数据文件的描述页和被分配使用的数据页。而在备份的过程中，如果数据库系统还在继续运行，这期间的数据库操作并不是都会立即体现到数据文件中，而是首先以日志的形式写到归档日志中，因此，为了保证用户可以通过备份集将数据恢复到备份结束时间点的状态，就需要将备份过程中产生的归档日志也保存到备份集中。

还原与恢复是备份的逆过程。还原是将备份集中的有效数据页重新写入目标数据文件的过程。恢复则是指通过重做归档日志，将数据库状态恢复到备份结束时的状态；也可以恢复到指定时间点和指定 LSN。恢复结束以后，数据库中可能存在处于未提交状态的活动事务，这些活动事务在恢复结束后的第一次数据库系统启动时，会由 DM 数据库自动进行回滚。备份、还原与恢复的关系如图 1.1 所示。

达梦数据库信创大赛资料

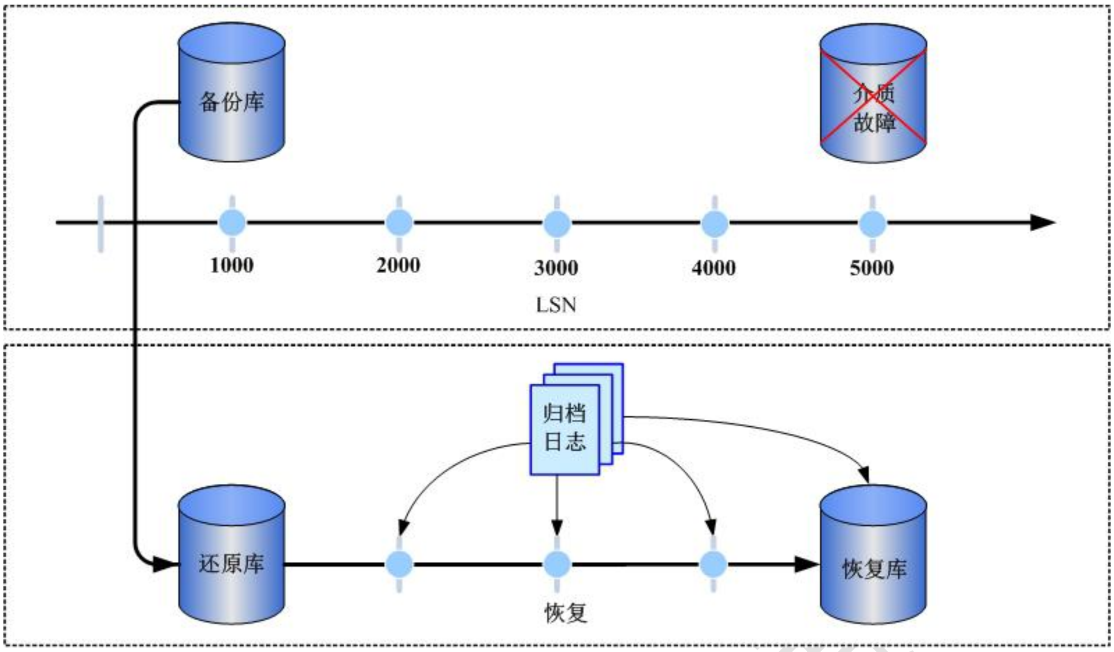


图 1.1 备份、还原与恢复

1.1基本概念

本节介绍与 DM 备份与还原相关的一些基本概念，了解这些基本概念是阅读后续章节的基础。

1.2.1 表空间与数据文件

DM 数据库的表空间是一个逻辑概念，其目的主要是为了方便数据库的管理，数据库的所有对象在逻辑上都存放在某个表空间中，而物理上都存储在所属表空间的数据文件中。一个表空间由一个或多个数据文件组成。

数据文件是数据库中最重要的文件类型，是真实数据存储的地方。DM 中数据文件的扩展名为.DBF，分为系统默认生成的数据文件和用户自己创建的数据文件两类。

在创建 DM 数据库时，系统会自动创建 5 个表空间：SYSTEM 表空间、ROLL 表空间、MAIN 表空间、TEMP 表空间和 HMAIN 表空间。DM 自动为这几个自动创建的表空间分别生成默认的数据文件：SYSTEM.DBF、ROLL.DBF、MAIN.DBF 和 TEMP.DBF，HMAIN 表空间没有默认的数据文件。

用户也可以创建自己的表空间，由用户创建的表空间统称为用户自定义表空间，在创建自定义表空间时需要为表空间指定数据文件。用户可通过为已存在的表空间增加数据文件或

创建一个新的表空间来创建数据文件。

因此，DM 中的表空间有以下几类：

SYSTEM 表空间

存放了 DM 数据库全局字典信息和全局系统数据，是 DM 数据库能够正常运行的必要前提，默认对应数据文件 SYSTEM.DBF。CREATE TABLE 等 DDL 操作会修改 SYSTEM 表空间数据。

ROLL 表空间

存放 DM 数据库运行过程中产生的所有回滚记录。DM 中几乎所有的数据库修改操作都会生成回滚记录，并保存在 ROLL 表空间的数据文件中。ROLL 表空间是数据库全局对象，不论修改哪一个表空间，生成的回滚记录都是写入 ROLL 表空间，该表空间由系统自动维护，默认数据文件为 ROLL.DBF。

TEMP 表空间

存放临时表数据以及数据库运行过程中产生的临时数据。在数据库运行过程中，SORT、HASH JOIN 等操作都可能会生成临时结果集，它们作为临时数据存放在 TEMP 表空间中。TEMP 表空间是数据库全局对象，由系统自动维护。若数据库重启，保存在 TEMP 表空间中的所有数据都会丢失。TEMP 表空间的默认数据文件为 TEMP.DBF。

DM 的物理备份还原不支持：TEMP 表空间。

MAIN 表空间

在创建用户时，如果没有指定默认表空间，系统自动指定 MAIN 表空间为用户默认的表空间。

HMAIN 表空间

DM 的 HUGE 表空间，用来存放 HUGE 表数据文件。

用户自定义表空间

即用户通过创建表空间相关操作生成的表空间。

1.2.2 重做日志

重做日志，又叫 REDO 日志，忠实记录了所有物理页的修改，基本信息包括操作类型、表空间号、文件号、页号、页内偏移、实际数据等。数据库中 INSERT、DELETE、UPDATE 等 DML 操作以及 CREATE TABLE 等 DDL 操作最终都会转化为对某些数据文件、某些数据页的修改。因此，在系统故障重启时，通过重做 REDO 日志，可以将数据库恢复到故障时的状

达梦数据库信创大赛资料

态。

DM 数据库默认包含两个扩展名为 log 的日志文件，用来保存 REDO 日志，称为联机重做日志文件。这两个文件循环使用。任何数据页从内存缓冲区写入磁盘之前，必须保证其对应的 REDO 日志已经写入到联机日志文件。

Redo 日志包（RLOG_PKG）是 DM 数据库保存 Redo 日志的数据单元，一个日志包内可保存一个或多个 PTX 产生的 Redo 日志。日志包具有自描述的特性，日志包大小不固定，采用固定包头和可变包头结合的方式，包头记录日志的控制信息，包括类型、长度、包序号、LSN 信息、产生日志的节点号、加密压缩信息、日志并行数等内容。

日志包生成时按照序号连续递增，相邻日志包的 LSN 顺序是总体递增的，但是在 DMDSC 集群环境下不一定连续。如果未开启并行日志，RLOG_PKG 包内日志的 LSN 是递增的。如果开启并行日志，一个 RLOG_PKG 包内包含多路并行产生的日志，每一路并行日志的 LSN 是递增的，但是各路之间的 LSN 并不能保证 LSN 有序，因此并行日志包内 LSN 具有局部有序，整体无序的特点。

1.2.3 归档日志

DM 数据库可以在归档和非归档两种模式下运行。DM 支持多种归档方式。系统在归档模式下运行会更安全，当出现介质故障，如磁盘损坏导致数据文件丢失、异常时，利用归档日志，系统可以恢复至故障发生的前一刻。因此，为了保证归档日志文件和数据文件不同时出现问题，建议将归档目录与数据文件配置、保存到不同的物理磁盘上。除了表备份还原，其他的联机备份与还原必须运行在归档模式下。

DM 数据库定义了多种归档方式，包括本地归档、实时归档、即时归档、异步归档和远程归档，其中本地归档和远程归档与备份还原密切相关。

系统将 REDO 日志先写入联机日志文件后，根据归档的配置情况，异步地将 REDO 日志写入本地归档日志文件，或者通过 MAL 系统发送到远程归档的目标实例，写入目标实例的远程归档日志文件中。

Primary/Normal 模式库归档日志文件的命名规则是：ARCH_NAME_DB_MAGIC[SEQNO]_日期时间.log。其中 ARCH_NAME 是在 dmarch.ini 中配置的 LOCAL/REMOTE 归档名称，DB_MAGIC 是生成日志的数据库魔数，SEQNO 代表 DSC 节点号，日期时间是归档日志文件的创建时间。比如：

达梦数据库信创大赛资料

ARCHIVE_LOCAL1_0x567891[0]_2018-05-30_10-35-34.log。

Standby 模式库（备库）归档日志文件的命名规则是：

STANDBY_ARCHIVE_DB_MAGIC[SEQNO]_日期时间.log 。其中，STANDBY_ARCHIVE 表示备库生成的归档日志文件；DB_MAGIC 是生成日志的数据库（主库）魔数；SEQNO 代表主库对应的 DSC 节点号；日期时间是归档日志文件的创建时间。比如：

STANDBY_ARCHIVE_0x123456[0]_2018-05-30_10-35-34.log 。由于 STANDBY_ARCHIVE 用于表示备库生成的归档日志，因此，不允许将归档名称配置为 STANDBY_ARCHIVE。

1.2.4 备份

备份的目的是当数据库遇到损坏的情况下，可以执行还原恢复操作，把数据库复原到损坏前的某个时间点。用于还原恢复数据库的载体是备份集，生成备份集的过程便是备份了。

备份就是从源库（备份库）中读取有效数据页、归档日志等相关信息，经过加密、压缩等处理后写入备份片，并将相关备份信息写入备份元数据文件的过程。

下面将根据数据组织形式、数据库状态、备份粒度、备份内容、以及备份的一致性和完整性等差异，对备份进行介绍。

1.2.8.1 逻辑备份和物理备份

逻辑备份是指利用 dexp 导出工具，将指定对象（库级、模式级、表级）的数据导出到文件的备份方式。逻辑备份针对的是数据内容，并不关心这些数据物理存储在什么位置。

物理备份则直接扫描数据库文件，找出那些已经分配、使用的数据页，拷贝并保存到备份集中。物理备份过程中，不关心数据页的具体内容是什么，也不关心数据页属于哪一张表，只是简单的根据数据库文件系统的描述，来挑选有效的数据页。

这两种备份方式，分别适应不同的应用场景，本文前三章介绍物理备份，后四章介绍逻辑备份。

1.2.8.2 联机备份和脱机备份

数据库处于运行状态、并正常提供数据库服务情况下进行的备份操作，我们称为联机备

达梦数据库信创大赛资料

份。数据库处于关闭状态时进行的备份操作，被称为脱机备份。

从 V2.0 版本开始，使用 DMRMAN 工具进行脱机备份，并且支持对异常关闭的数据库进行脱机库备份。备份异常关闭的数据库，要求配置了本地归档，如果本地归档不完整，则需要先修复本地归档，再进行备份。

联机备份则使用客户端工具连接到数据库实例后，通过执行 SQL 语句进行；也可以通过配置作业，定时完成自动备份。联机备份不影响数据库正常提供服务，是最常用的备份手段之一。

联机备份时，可能存在一些处于活动状态的事务正在执行，为确保备份数据的一致性，需要将备份期间产生的 REDO 日志一起备份。因此，只能在配置本地归档、并开启本地归档的数据库上执行联机备份。

注意：**只有已经关闭的数据库才允许执行脱机备份。正在运行的数据库，无法执行脱机备份，系统会报错。**

1.2.8.3 数据备份

数据备份主要针对数据文件内容，包括库备份、表空间备份和表备份。

库备份，顾名思义就是对整个数据库执行的备份，又称为库级备份。库备份会拷贝数据库中所有数据文件的有效数据页，如果是联机备份，则还会拷贝备份过程中产生的归档日志，写入到备份集中。

表空间备份是针对特定表空间执行的备份，又称为表空间级备份。表空间备份只能在联机状态下执行。

表备份则拷贝指定表的所有数据页到备份集中，并会记录各个数据页之间的逻辑关系用以恢复。表备份只能在联机状态下执行，一次表备份操作只能备份一张用户表，并且不支持增量表备份。

本章仅介绍库备份，其他备份方式详见达梦官方《DM8 备份和还原手册》。

1.2.8.4 完全备份和增量备份

按照备份数据完整性，可将备份分为完全备份和增量备份。库备份和表空间备份支持增量备份，表备份不支持增量备份。

达梦数据库信创大赛资料

完全备份生成的备份集包含了指定库（或者表空间）的全部有效数据页。当数据规模比较大的情况下，生成的完全备份集通常会比较大，而且备份时间也会比较长。

增量备份是在某个特定备份集基础上，收集数据库新修改的数据页进行备份，可以有效减少备份集的空间占用、提高备份速度。这个特定的、已经存在的备份集称为增量备份的基备份，根据对基备份的要求不同，DM 的增量备份分为以下两种：

1． 差异增量备份

差异增量备份的基备份既可以是一个完全备份集，也可以是一个增量备份集。

利用增量备份进行还原操作时，要求其基备份必须是完整的；如果差异增量备份的基备份本身也是一个增量备份，那么同样要求其基备份是完整的；任何一个增量备份，最终都是以一个完全备份作为其基备份。因此，完全备份是增量备份的基础。

2． 累积增量备份

累积增量备份的基备份只能是完全备份集，而不能是增量备份集。

增量备份的基备份集既可以是脱机备份生成的，也可以是联机备份生成的，脱机增量备份的基备份集可以是联机备份生成的，联机增量备份的基备份集也可以是脱机备份生成的。

1.2.5 还原与恢复

还原是备份的逆过程，就是从备份集中读取数据页，并将数据页写入到目标数据库对应数据文件相应位置的过程。

由于联机备份时，系统中可能存在一些处于活动状态的事务正在执行，并不能保证备份集中的所有数据页是处于一致性状态；而脱机备份时，数据页不一定是正常关闭的，也不能保证备份集中所有数据页是处于一致性状态。因此，还原结束后目标库有可能处于非一致性状态，不能马上提供数据库服务；必须要进行数据库恢复操作后，才能正常启动。

与备份类似，下面将从数据组织形式、数据库状态、还原粒度、还原内容等方面介绍还原的相关内容。

1.2.9.1 逻辑还原和物理还原

逻辑还原是逻辑备份的逆过程，逻辑还原就是使用 dimp 工具，把 dexp 导出的备份数据重新导入到目标数据库。

达梦数据库信创大赛资料

物理还原是物理备份的逆过程，物理还原一般通过 DMRMAN 工具（或者 SQL 语句），把备份集中的数据内容（数据文件、数据页、归档文件）重新拷贝、写入目标文件。

关于逻辑还原更详细的说明，参考本章逻辑导入导出章节。

1.2.9.2 联机还原和脱机还原

联机还原指数据库处于运行状态时，通过 SQL 语句执行还原操作。表还原可以在联机状态下执行。

脱机还原指数据库处于关闭状态时执行的还原操作，脱机还原通过 DMRMAN 工具进行。库备份、表空间备份和归档备份，可以执行脱机还原。脱机还原操作的目标库必须处于关闭状态。

1.2.9.3 数据还原

根据备份集类型，数据还原可以分为库还原、表空间还原和表还原。库还原和表空间必须脱机执行；表还原操作只能联机执行。

表空间还原的数据来源既可以是表空间备份集，也可以是库备份集。还原的目标表空间不能是 TEMP 表空间，只能是 MAIN、SYSTEM、ROLL 表空间，或者用户定义的表空间。

表还原从表备份集读取数据，重新恢复目标表数据，还会在目标表上重建索引、约束。

本章仅介绍库还原，其他还原方式详见达梦官方《DM8 备份和还原手册》。

1.2.9.4 完全还原和增量还原

完全还原是指直接利用完全备份集进行数据还原操作。增量还原指通过增量备份集进行数据还原操作。但是考虑到增量备份集的基础一定是一个完全备份集，因此增量还原过程中隐含了一个完全还原操作。如果增量备份集的基备份集被删除了，那么单独使用这个增量备份集是无法进行还原操作的。

达梦数据库信创大赛资料

1.2备份方式

当前仅支持备份集方式的备份还原，不再支持其他备份方式。备份还原实现策略有两种：dmap 辅助进程方式和无辅助进程方式。用户可通过 DM.INI 参数 bak_use_ap 来选择（dmrman 使用参数 use_ap），bak_use_ap 取值 1、2。默认为 1。

1：DMAP 辅助进程方式，可支持第三方备份（指定 DEVICE TYPE 为 TAPE）。DMAP 插件执行，改造了备份还原任务子系统，允许指定并行度，大幅提升了备份还原的效率，特别是加密、压缩的处理效率。如果选择使用 DMAP 辅助进程，执行备份还原之前就必须启动 DMAP 服务。安装 DM 数据库以后，DMAP 服务会自动启动。如果需要手动启动，有两种途径，一是启动 DM 服务查看器中的 DmAPService。二是通过手动启动 DMAP 执行码实现，DMAP 执行码位于 DM 安装目录的 bin 子目录下。除此之外，LINUX 下，还可以调用 bin 目录下的 DmAPService 脚本。

2：无辅助进程方式，不依赖 DMAP，由主进程 dmserver 自身执行备份还原，但不支持第三方备份（指定 DEVICE TYPE 为 TAPE）。

2 物理备份还原

本章介绍 DM 备份与还原的实现原理，读者通过阅读本章内容，可以了解 DM 备份、还原与恢复的实现原理。

2.1归档说明

备份与恢复过程都依赖归档日志，归档日志是保证数据一致性和完整性的重要保障。配有归档日志的数据库系统在出现故障时丢失数据的可能性更小，这是因为一旦出现介质故障如磁盘损坏时，利用归档日志，系统可被恢复至故障发生的前一刻，也可以还原到指定的时间点。

2.1.1 本地归档

Redo 日志本地归档（LOCAL），就是将 Redo 日志写入到本地归档日志文件的过程。

达梦数据库信创大赛资料

配置本地归档情况下，Redo 日志刷盘线程将 Redo 日志写入联机 Redo 日志文件后，对应的 RLOG_PKG 由专门的归档线程负责写入本地归档日志文件中。

与联机 Redo 日志文件可以被覆盖重用不同，本地归档日志文件不能被覆盖，写入其中的 Redo 日志信息会一直保留，直到用户主动删除；如果配置了归档日志空间上限，系统会自动删除最早生成的归档 Redo 日志文件，腾出空间。如果磁盘空间不足，且没有配置归档日志空间上限（或者配置的上限超过实际空间），系统将自动挂起，直到用户主动释放出足够的空间后继续运行。

DM 提供了按指定的时间或指定的 LSN 删除归档日志的系统函数（SF_ARCHIVELOG _DELETE_BEFORE_TIME 和 SF_ARCHIVELOG_DELETE_BEFORE_LSN），但需谨慎使用。避免归档日志缺失，导致数据无法恢复。

注意：

为了最大限度地保护数据，当磁盘空间不足导致归档写入失败时，系统会挂起等待，直到用户释放出足够的磁盘空间。

当磁盘损坏导致归档日志写入失败时，系统会强制 HALT。

2.1.2 归档切换

由于本地归档和远程归档是异步写入归档日志文件的，REDO 日志在写入联机日志文件后，再由专门的归档线程负责将这些 REDO 日志写入本地归档日志文件。通过归档切换功能，可以将这些已经写入联机日志文件，但还没有写入归档日志文件的 REDO 日志，写入到归档日志文件中。通过执行以下 SQL 命令，可以完成归档切换功能。三条语句功能一样，选择一条执行即可。

```
alter database archivelog current;
alter system archive log current;
alter system switch logfile;
```

2.2备份

任何一个对 DM 数据库的操作，归根结底都是对某个数据文件页的读写操作。物理备份就是把这些数据文件中的有效数据页备份起来，在出现故障时，用于恢复数据。DM 的物理备份一般包括数据备份和日志备份两部分，数据备份是拷贝数据页内容，日志备份则是拷贝备份过程中产生的 REDO 日志。

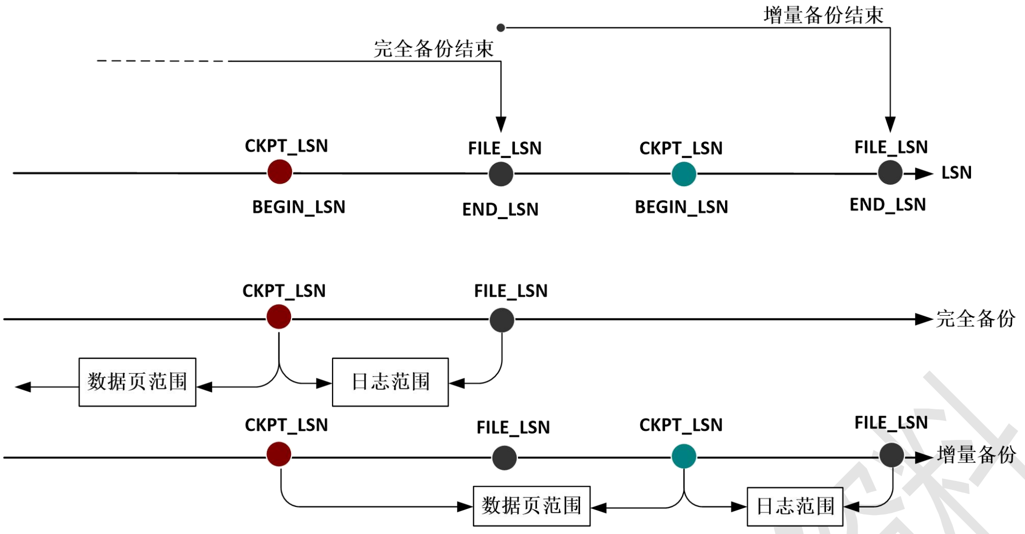


图 2.1 备份内容

2.2.1 数据备份

数据备份过程中，根据 DM 数据文件系统的描述信息，准确判断每一个数据页是否被分配、使用，将未使用的数据页剔除，仅保留有效数据页进行备份，这个过程我们称为智能抽取。与直接文件拷贝方式相比，DM 物理备份丢弃了那些没有使用的数据页，因此可以节省对存储空间的要求，并有效减少 IO 数量，提升备份、还原的效率。

■ **完全备份**

执行完全备份，备份程序会扫描数据文件，拷贝所有被分配、使用的数据页，写入到备份片文件中（如图 2.1 中第二根横轴所示）。库备份会扫描整个数据库的所有数据文件，表空间备份则只扫描表空间内的数据文件。

参照图 1.1，若执行库备份，则备份数据文件包括除 TEMP 表空间以外的其他表空间内的所有数据文件；若执行表空间备份，以用户表空间 TSDA 为例，那么只会扫描 TSDA 表空间的数据文件，包括 tsda.dbf、tsda_.dbf 和 tsda_b.dbf。数据文件备份结束后，BEGIN_LSN 之前修改的所有数据页都被备份下来。

■ **增量备份**

执行增量备份，备份程序会扫描数据文件，拷贝所有基备份结束以后被修改的数据页，写入到备份片文件中。

达梦数据库信创大赛资料

为了简化增量备份的还原过程，避免还原过程中重做基备份集对应的归档日志，DM 要求执行增量备份时，<=基备份 END_LSN 的所有数据页已经写入磁盘。也就是说增量备份会拷贝 LSN >基备份 BEGIN_LSN 的数据页写入备份片文件中（如图 2.1 中第三根横轴所示），LSN <= 基备份 BEGIN_LSN 的数据页不需要写入到备份片文件中。

同样库增量备份会扫描整个数据库的所有数据文件，表空间增量备份只扫描表空间内的数据文件。

2.3还原与恢复

还原与恢复是备份的逆过程，还原与恢复的主要目的是将目标数据库恢复到备份结束时刻的状态。还原的主要动作是将数据页从备份集中拷贝回数据库文件相应位置，恢复则是重做 REDO 日志将数据库恢复到一致性状态。

2.3.1 数据还原

2.3.1.1 库还原

库还原就是根据库备份集中记录的文件信息重新创建数据库文件，并将数据页重新拷贝到目标数据库的过程。DM 既可以将一个已存在的数据库作为还原目标库，也可以指定一个路径作为还原目标库的目录。库还原的主要步骤包括：清理目标库环境；重建数据库文件；拷贝数据页；重建联机日志文件；修改配置参数等。

- **清理目标库环境**

如果指定已存在的数据库作为还原目标库，还原操作首先解析 dm.ini 配置文件，获取 dm.ctl 控制文件路径，删除控制文件中的数据文件，然后根据 overwrite 选项，如果指定 overwrite 选项，若待还原文件存在，则删除；如果未指定 overwrite 选项，若待还原文件存在，则报错，但保留目标库的日志文件、控制文件等。

如果指定还原到一个目录，则根据 OVERWRITE 参数选择策略，检查目标目录内的 dm.ini 文件、dm.ctl 文件，默认的日志文件 DBNAME01.log 和 DBNAME02.log（其中 DBNAME 为数据库名称），待还原的数据文件等。如果用户指定 OVERWRITE，并且存在相关文件情况下，还原过程中会自动删除这些已经存在的文件；如果没有指定 OVERWRITE，并

且存在相关文件，则会报错。

■ **重建数据库文件**

如果将一个已存在数据库作为还原目标，则需要将目标数据库的 dm.ini 路径作为还原参数。还原过程中，会重新创建数据文件，并将相关信息写入 dm.ctl 控制文件中。

如果将数据库还原到指定目录，则会在这个目录创建一个 dm.ini 配置文件，设置 CTL_PATH、SYSTEM_PATH 配置项指向这个目录，并在这个目录下创建 dm.ctl 控制文件。DMDSC 不支持指定目录还原数据库。

数据文件重建策略：

1）目标库和备份集中的 SYSTEM_PATH 路径相同，则按照备份集中记录的原始路径创建文件；

2）目标库和备份集中的 SYSTEM_PATH 路径不相同，默认在 SYSTEM_PATH 目录创建文件；

3）如果已存在同名文件导致文件创建失败，会重命名文件后在 SYSTEM_PATH 目录创建，文件重命名规则是：DB_NAME+序号.DBF；如果重命名后仍然冲突，则会一直重试；

4）使用 mapped file 指定源文件与目标文件的映射关系，定制数据库文件的物理分布情况，可以很好的满足用户关于数据文件分布的需求。

■ **重建联机日志文件**

指定目录还原，系统目录使用指定还原目录，所有库配置文件均认为在指定还原目录下。联机日志文件命名规则，单机环境为 db_name+文件编号.log，其中 db_name 取自备份集备份库的名称，文件编号从 1 开始，如 DAMENG01.log、DAMENG02.log。联机日志文件至少 2 个。

■ **拷贝数据页**

拷贝数据页是从备份集中读取数据页，并将数据页写入数据文件指定位置的过程。由于备份过程中，只将有效的数据页写入备份集中，因此，还原过程也只涉及这些被分配使用的数据页。

■ **重置目标库**

具体包括：

达梦数据库信创大赛资料

- ✓ 更新日志信息，设置当前 CKPT_LSN 为备份集中 BEGIN_LSN，并设置日志文件状态为 INACTIVE；
- ✓ 更新 DB_MAGIC，还原后，库中 PERMANENT_MAGIC 仍与备份集中相同；
- ✓ 设置还原标志，标识当前库为指定库要还原的库，不允许使用；
- ✓ 更新目标库的控制文件 dm.ctl，把当前库中的数据文件信息都记录到控制文件中，使用备份集中的服务器秘钥文件，重新生成新的秘钥文件。

■ **修改配置参数**

还原到指定库时，默认会保留目标库的配置参数不变，也可以在还原时指定 REUSE DMINI 子句，使用备份集中的配置参数替换目标库 dm.ini 中的配置参数。还原到指定目录时，会重新建立一个 dm.ini 配置文件，并用备份集中的参数值来设置这些配置项。需要注意的是，一般与路径相关的配置参数，比如 SYSTEM_PATH 等并不会被替换，而是保留目标库 dm.ini 的原始值不变。

服务器秘钥文件(dm_service.prikey 或者 dm_external.config)仅在备份集中备份库非 usbkey 加密的情况下重建，并使用备份集中备份的秘钥内容进行还原。

注意事项：

1）指定的 dm.ini 必须存在且各项配置信息有效，其中 CTL_PATH 必须配置且路径必须有效；

2）若指定目录还原，则指定目录作为数据库系统目录处理；

3）由于还原是要确保数据库数据的完整性，因此，对于增量备份的还原，需要搜集完整的备份集链表，然后从前到后，逐个还原备份集中数据。鉴于增量备份 BEGIN_LSN 确定规则，增量备份的还原过程中，不需要重做任何归档日志。

2.3.2 数据恢复

数据恢复是指在还原执行结束后，重做 REDO 日志，将数据库恢复到一致性状态，并执行更新 DB_MAGIC 的过程。其中重做 REDO 日志可以多次执行，直到恢复到目标状态。还原结束后，必须经过恢复操作，数据库才允许启动。即使备份过程中没有修改任何数据，备份集不包含任何 REDO 日志，在数据库还原结束后，也必须使用 DMRMAN 工具执行数据恢复操

达梦数据库信创大赛资料

作后，才允许启动数据库。

数据恢复重做的 REDO 日志，既可以是那些在备份过程中产生的、包含在备份集中的 REDO 日志，也可以是备份数据库本地归档日志文件。在本地归档日志完整的情况下，数据还原结束后，可以利用本地归档日志，将数据库恢复到备份结束后任意时间点状态。

不管采用哪种数据恢复方法，REDO 日志的范围至少要覆盖备份过程中产生的 REDO 日志，也就是说必须完整包括备份集中记录的[BEGIN_LSN， END_LSN]之间的 REDO 日志，如果归档日志缺失将会导致数据库恢复失败。只有库备份和表空间备份还原后，需要执行数据恢复，表还原结束后，不需要执行数据恢复。

为了让读者更好地了解恢复的过程，这里先介绍一下 DM 的 MAGIC。PERMANENT_MAGIC 和 DB_MAGIC 是用来标识数据库的 INTEGER 类型值。DM 在初始化数据库时生成 PERMANENT_MAGIC 和 DB_MAGIC 值，其中 PERMANENT_MAGIC 一经生成，永远不会改变（DDL_CLONE 还原库的 PMNT_MAGIC 除外），称为数据库永久魔数。只有 DDL_CLONE 还原库的 PMNT_MAGIC 会发生改变，当一个库使用 DDL_CLONE 备份集还原并恢复之后，在执行 RECOVER DATABASE …… UPDATE DB_MAGIC 时，PMNT_MAGIC 会发生改变。DB_MAGIC 称为数据库魔数，同样可以用来表示某一个数据库，但 DB_MAGIC 是可以变化的，每经过一次还原、恢复操作后，DB_MAGIC 就会产生变化，用来区分备份源库和还原目标库。

可以通过下列语句查看系统的 PERMANENT_MAGIC 和 DB_MAGIC 值。

```
SELECT  PERMANENT_MAGIC;
SELECT  DB_MAGIC  FROM  V$RLOG;
```

2.3.2.1 指定备份集恢复

默认未指定 WITHOUT LOG 子句的联机库备份生成的备份集，包含了备份过程中产生的 REDO 日志，数据还原结束后，可以直接指定备份集，将数据库恢复到备份结束时的状态。

由于执行增量备份时，要求<=基备份 END_LSN 的所有数据页已经写入磁盘；因此基备份集中包含的 REDO 日志不需要重做，只要重做指定执行还原操作的备份集中包含的 REDO 日志，就可以将数据库恢复到一致性状态。

指定备份集恢复的简要过程包括：

1. 从备份集读取 REDO 日志，并生成一个临时的本地归档日志文件；
2. 利用生成的临时归档日志文件，重做 REDO 日志，并将数据修改写入磁盘；

达梦数据库信创大赛资料

3. 删除临时生成的归档日志文件；

4. 更新数据库日志信息，设置 CKPT_LSN 为最后一个重做的 REDO 日志 LSN 值；

5. 修改数据状态为 ACTIVE，标记数据库启动时需要进行相应的回滚活动事务、PURGE 已提交事务。

2.3.2.2 指定归档恢复

如果备份时指定了 WITHOUT LOG 子句，那么产生的备份集不包含备份过程中产生的 REDO 日志。这种备份集还原后，必须利用本地归档日志，将数据库恢复到一致性状态。执行恢复前，会检查本地归档日志文件的完整性，要求必须包括[BEGIN_LSN, END_LSN]之间完整的 REDO 日志。

利用本地归档日志进行恢复时，DMRMAN 工具会扫描指定的归档日志目录，收集与恢复数据库 PERMANENT_MAGIC 值相等的归档日志文件。与指定备份集恢复相比，利用本地归档日志恢复不需要生成、删除临时归档日志文件，其余的执行流程完全相同。

2.3.2.3 更新 DB_MAGIC

若备份集满足 BEGIN_LSN 等于 END_LSN ，即在备份过程中未产生 REDO 日志，则使用此备份集还原后只需要更新 DB_MAGIC 即可完成恢复。更新 DB_MAGIC 不重做 REDO 日志，仅仅更新库的 DB_MAGIC 值和数据库状态。

注意：**只能在还原后的数据库上执行更新 DB_MAGIC 操作。**

3 物理备份还原实战

本章详细介绍如何利用 DM 提供的各种工具进行备份还原与恢复的操作，包括 DIsql 工具、DMRMAN 工具、图形化客户端管理工具 MANAGER 和 CONSOLE。DIsql 工具用于执行联机的数据备份与数据还原，包括数据库备份、归档备份、表空间备份与还原、表备份与还原；DMRMAN 工具用于执行脱机的数据备份、还原与恢复，包括脱机的数据库备份、还原与恢复，

达梦数据库信创大赛资料

脱机还原表空间，归档的备份、还原与修复；客户端工具 MANAGER 和 CONSOLE 对应命令行工具 DIsql 和 DMRMAN 的功能，分别用于联机和脱机备份还原数据。这四种工具都可以独立使用，也可以相互配合，如使用 DIsql 或 MANAGER 工具联机备份的数据库备份文件可以用 DMRMAN 或 CONSOLE 工具还原。读者在阅读完本章的内容后，就可以动手配置归档，并进行备份与还原操作了。

本章内容主要包括：

- 准备工作
- 使用联机执行 SQL 语句进行备份还原
- 使用脱机工具 DMRMAN 进行备份还原
- 使用图形化客户端工具进行备份还原

警告：**OPEN 状态支持备份、还原和恢复的操作；**

MOUNT 状态支持归档备份；

SUPEND 状态均不支持。

3.1准备工作

本节介绍着手备份与还原之前的准备工作，主要内容包括：

- 支持与限制
- 归档配置

3.1.1 支持与限制

在着手备份与还原之前，先详细了解一下 DM 对备份与还原进行了哪些支持与限制，如下进行详细说明。

■ **联机备份**

对联机备份的支持与限制：

1） MPP 环境仅允许库和归档备份，且各节点都会执行，生成相应的备份集，支持 DDL CLONE；

2）DSC 环境支持库备份、表空间备份和表备份，要求 DSC 环境的所有节点都处于 OPEN 状态；

达梦数据库信创大赛资料

3） MOUNT 状态仅支持归档备份；

4） SUSPEND 状态所有备份均不支持；

5） OPEN 状态支持所有备份，支持 DDL CLONE；

6） PRIMARY 模式支持所有备份，支持 DDL CLONE；

7） STANDBY 模式仅支持库级、表空间级和归档备份，支持 DDL CLONE；

8） DDL CLONE 必须备份归档，不允许指定 WITHOUT LOG。

■ **联机还原：**

仅支持表级还原，对联机还原的支持与限制：

1） MPP 和分布式数据库不支持；

2） PRIMARY 支持；

3） SUSPEND 状态所有还原均不支持；

4） OPEN/NORMAL 支持。

■ **脱机备份**

脱机备份支持库级和归档备份。

1） MPP 视同单机环境，仅当前节点执行备份操作；

2） 允许异常退出后备份，支持 DDL_CLONE；

3） DSC 支持库级备份，支持 DDL_CLONE；当 DSC 环境下正常节点的 CKPT_LSN 小于故障节点的 CKPT_LSN 时，不支持脱机备份；

4） 分布式数据库不支持。

■ **脱机还原：**

脱机还原跟目标库所处的模式、状态以及集群环境（MPP 和 DSC）无关，允许库级、表空间级和归档还原。分布式数据库仅支持库级脱机还原恢复。

在一般的应用场景中，常规性的数据库维护工作，即在不影响数据库正常运行的情况下，建议定期执行联机数据库备份，且完全备份和增量备份结合使用。执行两次完全备份的时间间隔可以尽量长一点，在两次完全备份之间执行一定数量的增量备份，比如，可以选择每周执行一次完全备份，一周内每天执行一次增量备份。为了尽量减少对数据库正常工作的影响，建议备份时间，选择在工作量较少的时间，比如深夜。

达梦数据库信创大赛资料

说明：**无论管理员选用哪一种备份方式，都要注意不应将备份产生的备份集与源备份库存放在同一磁盘或同一存储介质上，以避免存储介质发生硬件故障时，源备份库与备份集同时被毁坏。**

注意：**备份与还原时，指定的备份集名称和目录名中最好不要包含中文、空格以及特殊字符，否则可能会因为字符处理及字符集问题导致一些不可预期的问题。**

3.1.2 归档配置

DM 数据库可以运行在归档模式或非归档模式下。如果是归档模式，联机日志文件中的内容保存到硬盘中，形成归档日志文件；如果是非归档模式，则不会形成归档日志。

采用归档模式会对系统的性能产生些许影响，然而系统在归档模式下运行会更安全，当出现故障时其丢失数据的可能性更小，这是因为一旦出现介质故障，如磁盘损坏时，利用归档日志，系统可被恢复至故障发生的前一刻，也可以还原到指定的时间点，而如果没有归档日志文件，则只能利用备份进行恢复。

通过 dm.ini 和 dmarch.ini 可以配置本地归档。dmarch.ini 生效的前提是 dm.ini 中的参数 ARCH_INI 置为 1。

DM 的 dmarch.ini 可以进行本地归档和远程归档的设置，DM 备份与还原过程中使用的日志均为本地归档日志。

dmarch.ini 中与备份还原相关的配置参数及其介绍见下表。

表 3. 1 dmarch.ini 相关配置项

配置项	配置含义
[ARCH_NAME]	REDO 日志归档名
ARCH_TYPE	REDO 日志归档类型，LOCAL 表示本地归档，REMOTE 表示远程
ARCH_DEST	REDO 日志归档目标，LOCAL 对应归档文件存放路径；REMOTE 对应远程目标节点实例名
ARCH_FILE_SIZE	单个 REDO 日志归档文件大小，取值范围（64M~2048M），缺省 1024M，即 1G

达梦数据库信创大赛资料

ARCH_SPACE_LIMIT	REDO 日志归档空间限制，当所有本地归档文件达到限制值时，系统自动删除最老的归档文件。0 表示无空间限制，取值范围（1024M~4294967294M），缺省为 0
ARCH_INCOMING_PATH	仅 REMOTE 归档有效，对应远程归档存放在本节点的实际路径

3.1.2.1 何时配置归档

联机备份数据库必须要配置归档。联机备份时，大量的事务处于活动状态，为确保备份数据的一致性，需要同时备份一段日志（备份期间产生的 REDO 日志），因此要求数据库必须配置本地归档且归档必须处于开启状态。

脱机备份数据库可配置归档也可以不配置。正常退出的库的备份不需要考虑本地归档日志的完整性，可以不配置归档；但对于故障退出的库的备份要求因故障未刷盘的日志也必须存在于本地归档中，因此必须配置归档，如果本地归档缺失，需要用户先修复归档，然后再备份。

备份表空间属于联机备份，必须配置归档。

备份表虽然是联机完全备份，但不需要配置归档。因为表在还原之后不需要再进行恢复操作，用不到归档日志。

备份归档日志必须配置归档。

3.1.2.2 配置本地归档

归档配置有两种方式：一是联机归档配置，数据库实例启动情况下，使用 SQL 语句完成 dmarch.ini 和 ARCH_INI 配置；二是手动配置归档，数据库实例未启动的情况下，手动编写 dmarch.ini 文件和设置参数 ARCH_INI。下面将分别说明这两种归档如何配置。

联机配置归档

使用 SQL 语句配置本地归档。

语法如下：

```
ALTER DATABASE <ADD|MODIFY|DELETE> ARCHIVELOG <归档配置语句>;
<归档配置语句>::= 'DEST = <归档目标>,TYPE = <归档类型>'
<归档类型>::=<local 方式>|<remote 方式>
```


达梦数据库信创大赛资料

```
<local 方式>::=LOCAL [,FILE_SIZE = <文件大小>][,SPACE_LIMIT = <空间大小限制>]
```

使用 SQL 语句开启和关闭归档模式。

语法如下：

```
ALTER DATABASE ARCHIVELOG | NOARCHIVELOG;
```

警告：

1.在归档模式下，不允许删除本地归档。

2.联机备份时，关闭已配置的本地归档之后再重新打开，会造成归档文件中部分日志缺失，备份时检查归档文件连续性时将会报错。存在该类操作时，若要避免该错误，备份前需要调用 checkpo int(100)主动刷新检查点。

例如，联机归档配置如下：

1)修改数据库为 MOUNT 状态。

```
SQL>ALTER DATABASE MOUNT;
```

2)配置本地归档。

```
SQL>ALTER DATABASE ADD ARCHIVELOG 'DEST = /home/dm_arch/arch, TYPE = local,
FILE_SIZE = 1024, SPACE_LIMIT = 2048';
```

3)开启归档模式。

```
SQL>ALTER DATABASE ARCHIVELOG;
```

4)修改数据库为 OPEN 状态。

```
SQL>ALTER DATABASE OPEN;
```

手动配置归档

1)手动编辑 dmarch.ini 文件，之后保存在 dm.ini 所在的目录。dmarch.ini 文件内容如下：

```
[ARCHIVE_LOCAL1]
ARCH_TYPE = LOCAL
ARCH_DEST = d:\dm_arch\arch
ARCH_FILE_SIZE = 1024
ARCH_SPACE_LIMIT = 2048
```

2)编辑 dm.ini 文件，设置参数 ARCH_INI=1，保存。

3)启动数据库实例，数据库已运行于归档模式。

说明：多路归档，指配置多个本地归档。配置的第一个归档，称为第一路归档，后面依次是第二路、第三路......

3.2使用联机执行 SQL 语句进行备份还原

本节介绍使用 DIsql 工具如何实现数据文件的备份及还原。

DM 支持通过联机执行 SQL 语句方式对数据库执行备份还原操作。联机方式支持数据库、用户表空间、用户表和归档的备份，用户表的还原。在进行联机库级备份、归档备份和表空间备份时，必须保证系统处于归档模式，否则联机备份不能进行。

需要说明的是，DM 新备份还原在执行联机操作时，语句的解析执行使用的是 DM 服务器的编码方式，而文件操作使用操作系统的编码方式。因此，当 DM 建库参数的编码方式与操作系统编码方式不一致时，使用中文文件名或路径名等可能造成控制台打印信息和日志文件中的信息的中文部分显示乱码。

3.2.1 数据库备份

在 DIsql 工具中使用 BACKUP 语句你可以备份整个数据库。通常情况下，在数据库实例配置归档后输入以下语句即可备份数据库：

```
SQL>BACKUP DATABASE BACKUPSET 'db_bak_01';
```

语句执行完后会在默认的备份路径下生成名为"db_bak_01"的备份集目录，默认的备份路径为 dm.ini 中 BAK_PATH 配置的路径，若未配置，则使用 SYSTEM_PATH 下的 bak 目录。这是最简单的数据库备份语句，如果要设置其他的备份选项需了解联机备份数据库的语法。语法如下：

```
BACKUP DATABASE [[[FULL] [DDL_CLONE]]| INCREMENT [CUMULATIVE][WITH BACKUPDIR '<基备份搜索目录>'{,'<基备份搜索目录>'} |[BASE ON <BACKUPSET '<基备份目录>']][TO <备份名>][BACKUPSET '<备份集路径>']
[BACKUPINFO '<备份描述>']];
```

FULL：备份类型。FULL 表示完全备份，可不指定，默认为完全备份。

INCREMENT：备份类型。INCREMENT 表示增量备份，若要执行增量备份必须指定该参数。

CUMULATIVE：用于增量备份中，指明为累积增量备份类型，若不指定则缺省为差异增量备份类型。

WITH BACKUPDIR：用于增量备份中，指定基备份的搜索目录，最大长度为256个字节。若不指定，自动在默认备份目录下搜索基备份。如果基备份不在默认的备份目录下，增量备份必须指定该参数。

BASE ON：用于增量备份中，指定基备份集目录。

TO：指定生成备份名称。若未指定，系统随机生成，默认备份名格式为：DB_库名_备份类型_备份时间。其中，备份时间为开始备份时的系统时间。

BACKUPSET：指定当前备份集生成路径。若指定为相对路径，则在默认备份路径中生成备份集。若不指定，则在默认备份路径中按约定规则，生成默认备份集目录。库级备份默认备份集目录名生成规则：DB_库名_备份类型_备份时间，如DB_DAMENG_FULL_20180518_143057_123456。

BACKUPINFO：备份的描述信息。最大不超过256个字节。

使用说明：

1． 备份成功后会在<备份集路径>或者备份默认目录下生成备份集。备份集中包括一个备份元数据文件，后缀.meta，一个或多个备份片文件，后缀.bak。

2． 当备份数据超过限制大小时，会生成新的备份文件，新的备份文件名是初始文件名后加文件编号。

3． 系统处于归档模式下时，才允许进行数据库联机备份。

4． MOUNT状态下不允许进行数据库备份。

完全备份和增量备份是数据库备份的最常用策略，本节将详细介绍这两种类型备份。

完全备份

执行数据库备份，数据库必须处于OPEN状态，MOUNT和SUSPEND状态下不允许执行数据库备份。

数据库完全备份中包含了指定库的全部有效数据页，为了保证数据安全应该对数据库定期执行完全备份。多久执行一次完全备份才合适，一般规则是数据库经过频繁修改后最好执行一次完全备份。

完全备份数据库步骤如下：

1） 保证数据库处于OPEN状态和归档开启状态。

2） DIsql 中输入备份数据库语句，最简单的不设置其他参数的完全备份语句如下：

```
SQL>BACKUP DATABASE FULL BACKUPSET '/home/dm_bak/db_full_bak_01';
```

备份语句中的参数 FULL 参数可以省略，不指定备份类型会默认指定备份类型为完全备份。

增量备份

增量备份指基于指定的库（或者表空间）的某个备份（完全备份或者增量备份），备份自该备份以来所有发生修改了的数据页。执行增量备份的主要目的是快速备份数据库中的修改，减少备份时间和避免重复的备份。

如何制定备份策略要根据可接受的最小恢复时间。例如，每周执行一次完全备份，每天执行一次增量备份，那么恢复时要重做的归档就不会超过一天。

增量备份数据库步骤如下：

1） 配置归档，保证数据库处于 OPEN 状态。

2） DIsql 中输入备份数据库语句，最简单的不设置其他参数的增量备份语句如下：

```
SQL>BACKUP DATABASE INCREMENT WITH BACKUPDIR '/home/dm_bak'BACKUPSET
'/home/dm_bak/db_increment_bak_02';
```

备份语句中的 INCREMENT 参数不可省略，该参数用来指定执行的备份类型为增量备份。若要创建累积增量备份，还需要指定 CUMULATIVE 参数，否则缺省为差异增量备份。WITH BACKUPDIR 参数用来指定基备份集的搜索目录，如果基备份集不在默认目录该参数不可省略。

3.3 使用脱机工具 DMRMAN 进行备份还原

DMRMAN（DM RECOVERY MANEGER）是 DM 的脱机备份还原管理工具，由它来统一负责库级脱机备份、脱机还原、脱机恢复等相关操作，该工具支持命令行指定参数方式和控制台交互方式执行，降低了用户的操作难度。

通过 DMRMAN 工具执行脱机操作过程中，仅会使用通过关键字 DATABASE 指定的目标库的本地归档配置信息，不会对本地归档配置文件中其他类型归档配置信息进行校验。例如：DATABASE 指定目标库为 /opt/dmdbms/data/dm.ini，为单机环境且配置参数 ARCH_INI 为 1，那么只会相应的读取 dmarch.ini 中本地归档日志信息。此时，即使 dmarch.ini 中配置了 REALTIME、MARCH、TIMELY 等其他类型的归档日志，也不会去校

达梦数据库信创大赛资料

验 dmmal.ini、dmtimer.ini 等是否有效。

本章节内容主要包括：

- DMRMAN 概述
- 启动 DMRMAN
- 数据备份
- 数据库还原和恢复
- 查看操作日志

3.3.1 DMRMAN 概述

DMRMAN 是 DM 提供的命令行工具，无需额外地安装。DMRMAN 的结构比较复杂，为了更好地使用它我们需要了解它的结构。接下来将对 DMRMAN 的重要组成部分进行介绍。

源库

源库是待备份的数据库。使用 DMRMAN 工具只可对数据库进行脱机备份，联机备份需要通过 DIsql 工具实现。DMRMAN 可以针对整个数据库执行脱机完全备份和增量备份，数据库可以配置归档也可以不配置。

目标库

目标库是待还原的数据库，即是用来做还原的库，也称为目标还原库。目标库可以是通过 dm.ini 文件指定的数据库，也可以是目标文件目录。通过 dm.ini 指定库时，dm.ini 及其配置中的 CTL_PATH 必须有效，且库必须处于关闭状态；指定文件目录时，指定目录作为 SYSTEM_PATH 处理，可以存在，也可以不存在，但必须有效。目前目标库的还原仅支持脱机还原，可通过 DMRMAN 和 CONSOLE 工具实现。

DMRMAN 客户端

DMRMAN 客户端是 DM 软件的一部分，用来执行目标数据库的备份和恢复操作。DMRMAN 客户端是一个命令行工具，命令行的好处是管理员可以编写复杂的脚本，将 DMRMAN 和操作系统的任务调度结合起来可以实现备份的自动化。

备份集

利用 DMRMAN 工具将数据库中的一个或多个数据文件、数据库信息等备份到一个称为“备份集”的逻辑结构中，备份集的格式是特定的只能由 DMRMAN 创建和访问。一个备份集中会包含一个或多个二进制文件，这个二进制文件被称为备份片。备份数据库会产生一个备份集，

达梦数据库信创大赛资料

这个备份集可能包含一个或者多个备份片。备份片的个数由数据库中数据文件的大小和文件系统对文件大小的限制共同决定。如果备份的数据文件大小超过了文件系统对文件大小的限制就会产生多个备份片。

在对数据库进行备份时可以指定备份片的大小来适应存储设备的容量。例如，一个数据库的产生的备份集大小约为 40GB，一盘磁带的容量为 10GB，这时可以指定备份片大小为 10G，备份后备份集将包含 4 个备份片，正好使用 4 盘磁带。

3.3.2 启动 DMRMAN

安装 DM 数据库后，DMRMAN 可执行程序与数据库其他可执行程序一样位于安装路径的执行码目录下。比如，LINUX 上数据库的执行码目录为/opt/dmdbms/bin，转到执行码目录直接在操作系统的命令行中输入以下命令就可启动 DMRMAN。若配置了环境变量 DM_HOME，可直接命令行执行：

```
./dmrman
```

启动后控制台中输入 exit 命令即可退出 DMRMAN 环境：

```
RMAN>exit
```

3.3.3 数据备份

本节介绍使用 DMRMAN 工具如何执行数据库和归档的备份操作。

3.3.3.1 备份数据库

本节主要描述使用 DMRMAN 如何执行基本的脱机数据库备份及实施一些备份策略，包括：

- 创建完全备份
- 创建增量备份

创建完全备份

执行数据库备份要求数据库处于脱机状态。与联机备份数据库不同的是，脱机备份不需要配置归档。一个完整的创建脱机数据库备份的示例如下：

1） 启动 DMRMAN 命令行工具。

2） 保证数据库处于脱机状态。

3） DMRMAN 中输入以下命令：

```
RMAN>BACKUP DATABASE '/opt/dmdbms/data/DAMENG/dm.ini' FULL BACKUPSET
'/home/dm_bak/db_full_bak_01';
```

命令中的 FULL 参数表示执行的备份为完全备份，也可以不指定该参数，DMRMAN 默认执行的备份类型为完全备份。

创建增量备份

增量备份指基于指定的库的某个备份（完全备份或者增量备份），备份自该备份以来所有发生修改了的数据页。脱机增量备份要求两次备份之间数据库必须有操作，否则备份会报错。

增量备份示例如下：

1） 启动 DMRMAN 命令行工具。

2） 保证数据库处于脱机状态。

3） DMRMAN 中输入以下命令：

```
RMAN>BACKUP DATABASE '/opt/dmdbms/data/DAMENG/dm.ini' INCREMENT WITH BACKUPDIR
'/home/dm_bak'BACKUPSET '/home/dm_bak/db_increment_bak_02';
```

命令中的 INCREMENT 参数表示执行的备份为增量备份，增量备份该参数不可省略。如果增量备份的基备份不在默认备份目录，必须指定 WITH BACKUPDIR 参数用于搜索基备份集。

3.3.4 数据库还原和恢复

本章节介绍在数据文件损坏后如何使用 DMRMAN 还原和恢复数据库，主要内容包括：

- 数据库还原
- 数据库恢复

3.3.5.1 执行数据库还原和恢复

3.3.5.1.1 数据库还原

使用 RESTORE 命令完成脱机还原操作，在还原语句中指定库级备份集，可以是脱机库级备份集，或是联机库级备份集。数据库的还原包括数据库配置文件还原和数据文件还原，目前可能需要还原的数据库配置文件包括 dm.ini、dm.ctl、服务器秘钥文件(dm_service.prikey 或者 dm_external.config、联机日志文件。

达梦数据库信创大赛资料

语法如下：

```
RESTORE DATABASE <restore_type>[WITH CHECK] FROM BACKUPSET '<备份集目录>'
[DEVICE TYPE DISK|TAPE[PARMS '<介质参数>']]
[WITH BACKUPDIR '<基备份集搜索目录>'{,'<基备份集搜索目录>'}]
[RENAME TO '<数据库名>'];
<restore_type>::=<type1>|<type2>
<type1>::='<INI 文件路径>'[REUSE DMINI][OVERWRITE] [FORCE]
<type2>::= TO '<system_dbf 所在路径>' [OVERWRITE]
```

DATABASE：指定还原库目标的 dm.ini 文件路径。

WITH CHECK：指定还原前校验备份集数据完整性。缺省不校验。

BACKUPSET：指定用于还原目标数据库的备份集目录。若指定为相对路径，会在默认备份目录下搜索备份集。

WITH BACKUPDIR：指定备份集搜索目录。

TASK THREAD：指定还原过程中用于处理解压缩和解密任务的线程个数。若未指定，则默认为 4；若指定为 0，则调整为 1；若指定超过当前系统主机核数，则调整为主机核数。

RENAME TO：指定还原数据库后是否更改库的名字，指定时将还原后的库改为指定的数据库名，默认使用备份集中的 db_name 作为还原后库的名称。

OVERWRITE：还原数据库时，存在重名的数据文件时，是否覆盖重建，不指定时默认报错。包含 WITH HUGE 选项产生的备份集，还原时，如果目标库路径下存在同名的表空间目录，将删除同名目录并重建，否则保错。

警告：**还原时指定 OVERWRITE 选项，所有重名的文件和非空目录均会删除重建，该过程是无法撤销的。为避免删除重要的文件，还原前务必确保数据库系统路径下未存放无关文件。**

<restore_type>：

■ <type1>

1>指定 dm.ini 还原，要求 dm.ini 中 CTL_PATH 必须配置正确，且内容有效；若配置 CTL_PATH 文件故障，且想利用 dm.ini 优化配置，则可选择 type2 还原后，用指定 dm.ini 覆盖还原后 dm.ini，并修改 CTL_PATH 为当前控制文件路径即可；

2>除 dm.ini 文件外，其他文件均可不存在；但 dm.ini 参数配置必须正确，且配置的 dm.ctl 文件必须是有效的控制文件；

3>数据库配置文件中除已经存在的 dm.ini 外，先删除控制文件中的数据文件，然后根据 overwrite 选项，若指定 overwrite，其他文件（这些文件不在控制文件中，所以未删除）均采用删除重建的处理，避免存在非法的文件，否则如果这些文件已经存在，则报错；

4>若指定 REUSE DMINI,则会将备份集中备份的 dm.ini 中除路径相关的 INI 参数外，均拷贝到当前 dm.ini 上。

5>若 dm.ini 记录的目标环境是 DMTDD 前端库，且目标环境不满足备份集的副本数和区块策略条件，则可以指定 FORCE 选项来强制还原，此时强制重建数据文件的策略如下：

对非 DMTDD 的备份集，优先使用 dmdfs.ini 配置的副本数重建文件，且系统表空间默认使用微区，用户定义表空间默认使用宏区。

若目标环境的可用副本数不足，（也可能为 0，即没有配置宏区或微区，此时需要调整区块策略），则按可用副本数重建文件。

■ <type2>说明：

1>所有文件均可不在，system_dbf 所在路径需为有效路径，若不存在，restore 过程中会自动创建；

2>所有前面提到到数据库配置文件均会在指定的 system_dbf 所在路径还原，但非单机环境中相关其他文件均不修改或者重建，如 MPP 中 dmmpp.ini、dmmal.ini 等；

3>若未指定 OVERWRITE，若 system_dbf 所在路径中存在待还原的库配置文件，则报错；若指定，则将已经存在的文件删除重建；

■ 联机日志

上述<type1>和<type2>中的还原后的联机日志文件至少会有两个，因为源库中的日志文件可能大于等于两个，如果小于两个（被误删的情况）的则补齐为两个。已经存在的联机日志配置，使用原路径，若文件大小非法，则使用缺省大小 256M 重建；缺少的使用缺省命名和缺省大小 256M 重建。联机日志文件的命名规则：

1>单机：db_name+(file_id+1).log，其中 fil_id + 1，占 2 个位置，如：db_name=DAMENG，则相应的名称为 DAMENG01.log，DAMENG02.log；

使用说明：

通过 RESTORE 命令还原后的数据库不可用，需进一步执行 RECOVER 命令，将数据库恢

达梦数据库信创大赛资料

复到备份结束时的状态。

数据库备份集分为联机和脱机两种类型。通常情况下，用户会在联机的情况下备份数据库，因此下面以联机数据库备份为例说明使用 DMRMAN 如何执行数据库还原操作。

1） 联机备份数据库，保证数据库运行在归档模式及 OPEN 状态。

```
SQL>BACKUP DATABASE BACKUPSET '/home/dm_bak/db_full_bak_for_restore';
```

2） 准备目标库。还原目标库可以是已经存在的数据库，也可使用 dminit 工具初始化一个新库。如下所示：

```
./dminit path=/opt/dmdbms/data db_name=DAMENG_FOR_RESTORE
```

3） 校验备份，校验待还原备份集的合法性。校验备份有两种方式，联机和脱机，此处使用脱机校验。

```
RMAN>CHECK BACKUPSET '/home/dm_bak/db_full_bak_for_restore';
```

4） 还原数据库。启动 DMRMAN，输入以下命令：

```
RMAN>RESTORE DATABASE '/opt/dmdbms/data/DAMENG_FOR_RESTORE/dm.ini' FROM
BACKUPSET '/home/dm_bak/db_full_bak_for_restore';
```

小窍门：**可通过调整 dm.ini 中的检查点和 REDO 日志相关参数，降低检查点频率，增大 REDO 日志包大小来提升还原性能。**

3.3.5.1.2 数据库恢复

使用 RECOVER 命令完成数据库恢复工作，可以是基于备份集的恢复工作，也可以是使用本地归档日志的恢复工作。如果还原后，数据已经处于一致性状态了，则可以使用更新 DB_MAGIC 方式恢复，前提是不需要重做日志。

语法如下：

```
RECOVER DATABASE '<INI 文件路径>'[FORCE] <with_archdir_lst_stmt>
[USE DB_MAGIC <db_magic>] [UNTIL TIME '<时间串>'] [UNTIL LSN <LSN>]; |
RECOVER DATABASE '<INI 文件路径>' [FORCE] FROM BACKUPSET '<备份集目录>'[DEVICE TYPE
DISK|TAPE[PARMS '<介质参数>']];
<with_archdir_lst_stmt> ::=
WITH ARCHIVEDIR '<归档日志目录>'{,'<归档日志目录>'}
```

达梦数据库信创大赛资料

DATABASE：指定还原库目标的 dm.ini 文件路径。

USE DB_MAGIC：指定本地归档日志对应数据库的 DB_MAGIC，若不指定，则默认使用目标恢复数据库的 DB_MAGIC。

FORCE：若恢复到 DMTDD 前端库，且恢复的 REDO 中包含创建表空间的记录，则可以指定该选项来强制创建表空间而忽略 REDO 中对该表空间的副本数和区块策略的严格限制。具体用法参考 3.3.5.1.1 数据库还原章节中 RESTORE 命令的 FORCE 选项。

UNTIL TIME：恢复数据库到指定的时间点。如果指定的结束时间早于备份结束时间，忽略 UNTIL_TIME 参数，重做所有小于备份结束 LSN（END_LSN）的 REDO 日志，将系统恢复到备份结束时点状态。

UNTIL LSN：恢复数据库到指定的 LSN。如果指定的 UNTIL_LSN 小于备份结束 LSN（END_LSN），则报错：恢复过程指定 UNTIL LSN***小于备份结束 LSN***。

BACKUPSET：指定用于还原目标数据库的备份集目录。

WITH ARCHIVEDIR：本地归档日志搜索目录。

数据库恢复是指重做 REDO 日志，有两种方式：从备份集恢复，即重做备份集中的 REDO 日志；从归档恢复，即重做归档中的 REDO 日志。数据库恢复后，还需要执行数据库更新操作，将数据库调整为可正常工作的库才算完成。下面逐一进行说明。

从备份集恢复

如果备份集在备份过程中生成了日志，且这些日志在备份集中有完整备份（如联机数据库备份），在执行数据库还原后，可以重做备份集中备份的日志，将数据库恢复到备份时的状态，即从备份集恢复。完整的示例如下：

1） 启动 DIsql 联机备份数据库。

```
SQL>BACKUP DATABASE BACKUPSET '/home/dm_bak/db_full_bak_for_recover_backupset';
```

2） 准备目标库，可以使用备份库，也可以重新生成库。重新生成库操作如下：

```
./dminit path=/opt/dmdbms/data db_name=DAMENG_FOR_RESTORE auto_overwrite=1
```

3） 启动 RMAN，校验备份。

```
RMAN>CHECK BACKUPSET '/home/dm_bak/db_full_bak_for_recover_backupset';
```

4） 还原数据库。

```
RMAN>RESTORE DATABASE '/opt/dmdbms/data/DAMENG_FOR_RESTORE/dm.ini' FROM
```

达梦数据库信创大赛资料

```
BACKUPSET '/home/dm_bak/db_full_bak_for_recover_backupset';
```

5） 恢复数据库。

```
RMAN>RECOVER DATABASE '/opt/dmdbms/data/DAMENG_FOR_RESTORE/dm.ini' FROM
BACKUPSET '/home/dm_bak/db_full_bak_for_recover_backupset';
```

从归档恢复

从归档恢复是利用重做本地归档日志来恢复数据的过程。从归档恢复允许恢复到指定的时间点及指定的 LSN 值。若同时指定了时间点和 LSN，则以较早的为结束点。

可以使用从归档恢复的情况：

1、执行过从备份集还原，以及执行过从备份集还原恢复的库，本地归档日志属于备份集备份时的库，本地归档日志的 DB_MAGIC 与备份集中记录 DB_MAGIC 值；

2、未执行过还原的库，但恢复目标库与本地归档日志来源于同一个库，且目标库与原来的库分离后没有再单独执行过操作或者故障重启过。比如一个数据库 A 因故障关闭后，拷贝一份作为数据库 B，此时数据库 A 和 B 完全相同。此后，数据库 A 故障重启，并正常执行其他操作，生成了新的本地归档日志。如果想利用 A 的本地归档日志去将 B 恢复到 A 的状态，那么可以利用 DMRMAN 工具执行从本地归档恢复。若中间 B 启动过或者执行过其他操作，则均不能再使用 A 的归档日志进行恢复操作。

利用归档恢复数据库至最新状态的完整示例如下：

1） 启动 DIsql 联机备份数据库，以及备份从检查点开始的本地归档日志。

```
SQL>BACKUP ARCHIVELOG FROM LSN 421401 BACKUPSET
'/home/dm_bak/arch_bak_lsn_421401';
SQL>BACKUP DATABASE BACKUPSET '/home/dm_bak/db_full_bak_for_recover_arch';
```

小窍门：**查看动态视图 V$RLOG 中的 CKPT_LSN 列，可以得出当前检查点 LSN。如：**

```
SQL> select CKPT_LSN from V$RLOG;

行号         CKPT_LSN
----------  --------------------
1           421401
```

2） 准备目标库，可以使用备份库，也可以重新生成库。

如果使用新生成的库，生成之后需要先重启一下数据库实例，才可以被还原。重新生成库操作如下：

```
./dminit path=/opt/dmdbms/data db_name=DAMENG_FOR_RESTORE auto_overwrite=1
```

3） 启动 RMAN，校验备份。

```
RMAN>CHECK BACKUPSET '/home/dm_bak/db_full_bak_for_recover_arch';
```

4） 还原数据库。

```
RMAN>RESTORE DATABASE '/opt/dmdbms/data/DAMENG_FOR_RESTORE/dm.ini' FROM
BACKUPSET '/home/dm_bak/db_full_bak_for_recover_arch';
```

3.3.5.1.3 数据库更新

数据库更新是指更新数据库的 DB_MAGIC，并将数据库调整为可正常工作状态，与数据库恢复一样使用 RECOVER 命令完成。数据库更新发生在重做 REDO 日志恢复数据库后，或者目标库不需要执行重做日志已经处于一致状态的情况。

语法如下：

```
RECOVER DATABASE '<INI 文件路径>' UPDATE DB_MAGIC;
```

DATABASE：指定还原库目标的 dm.ini 文件路径。

以使用正常退出数据库的脱机备份的备份集还原为完整示例，进行说明如下：

1） 启动 RMAN 备份数据库，保证数据库处于正常退出的脱机状态。

```
RMAN>BACKUP DATABASE '/opt/dmdbms/data/DAMENG/dm.ini' BACKUPSET
'/home/dm_bak/db_full_bak_for_recover_dbmagic';
```

2） 准备目标库，可以使用备份库，也可以重新生成库或者直接指定目录还原。重新生成库操作如下：

```
./dminit path=/opt/dmdbms/data db_name=DAMENG_FOR_RESTORE auto_overwrite=1
```

3） 校验备份。

```
RMAN>CHECK BACKUPSET '/home/dm_bak/db_full_bak_for_recover_dbmagic';
```

4） 还原数据库。

```
RMAN>RESTORE DATABASE '/opt/dmdbms/data/DAMENG_FOR_RESTORE/dm.ini' FROM
BACKUPSET '/home/dm_bak/db_full_bak_for_recover_dbmagic ';
```

5） 恢复数据库。

```
RMAN>RECOVER DATABASE '/opt/dmdbms/data/DAMENG_FOR_RESTORE/dm.ini' FROM
BACKUPSET '/home/dm_bak/db_full_bak_for_recover_dbmagic ';
```

6） 更新数据库。

达梦数据库信创大赛资料

```
RMAN>RECOVER DATABASE '/opt/dmdbms/data/DAMENG_FOR_RESTORE/dm.ini' UPDATE
DB_MAGIC;
```

3.3.5 查看操作日志

本节主要介绍整个操作过程中可能涉及的日志或跟踪文件。默认情况下，这些文件都位于 DM 数据库的 log 目录中。文件名中的“XXX”为生成文件的年月，如 2018 年 5 月生成日志文件其文件名中的“XXX”部分就是“201805”，每个月生成一个日志文件，在该月中日志内容都是不断增加的。操作日志类型包括：

- 全局日志文件 dm_DMSERVER_xxx.log
- 备份还原日志文件 dm_BAKRES__xxx.log

全局日志文件

dm_DMSERVER_xxx.log 是全局日志文件，全局日志文件不可配置。dm_DMSERVER_xxx.log 是 DM 数据库的运行日志，数据库运行过程中的日志信息都包含在这个日志中，备份还原过程中备份还原操作的阶段性信息会记录在这个日志文件中，包括日志生成时间、日志级别、日志生成进程名称和进程 ID、日志生成线程和内容等。

日志格式：时间+日志类型（INFO/WARN/ERROR/FATAL）+进程(database)+进程 ID(Pxxxx)+线程(dm_sql_thd/main_thread 等)+日志内容。

备份还原日志文件

dm_BAKRS_xxx.log 是备份还原日志文件，备份还原日志文件不可配置。dm_BAKRS_xxx.log 是 DM 备份还原在执行备份与还原时生成的日志，包括主进程（联机时为 dmserver 工作线程，脱机时为 dmrman）发送，以及 AP 进程处理过程信息。每条信息，包括生成时间、信息级别、生成日志的进程名称和进程 ID，对于 AP 进程还会生成对应主进程的 ID。另外，为了保证日志内容的完整性，还会记录由全局日志生成接口生成的日志记录，这部分日志格式参照全局日志文件说明。

备份还原日志格式为：

时间+日志类型（INFO/WARN/ERROR/FATAL/CMD/CMD_PARSE）+进程

达梦数据库信创大赛资料

（dmrman/database/dmbakres）+进程 ID(P 开头)+主进程 ID(PP 开头，非 dmbakres 显示 4294967295，非法值，否则为相应 dmrman 或者 database 的进程号)+日志内容。

3.4使用图形化客户端工具进行备份还原

DM 数据库提供几种类型的图形化客户端管理工具。在 WINDOWS 的 DM 数据库系统，可以通过选择"开始"-"程序"-"达梦数据库"-"客户端"，然后选择具体的某个客户端工具。比如，通过 MANAGER 管理工具可以与数据库进行交互，操作数据库对象和从数据库获取信息；通过控制台工具 CONSOLE 可以完成数据库的配置与监控等。

下文分别介绍使用MANAGER管理工具联机执行备份还原和使用CONSOLE控制台工具脱机执行备份还原。

3.4.1 使用 MANAGER 工具进行联机备份还原

下面将介绍如何使用 DM 的 MANAGER 管理工具来执行联机的备份与还原的操作。MANAGER 管理工具主界面如图 3.1 所示：

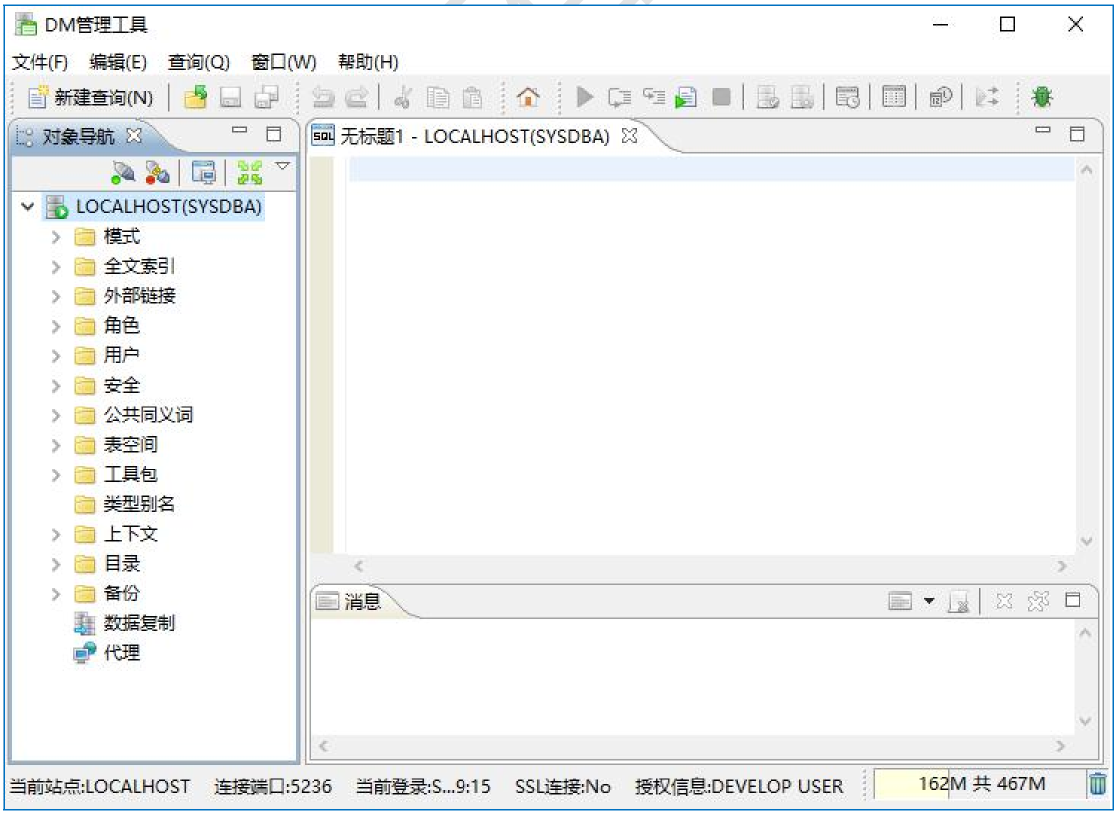


图 3.1 MANAGER 工具主页面

MANAGER 工具的对象导航树中的备份节点提供了联机备份还原相关的各项操作。导航树备份节点下面包含三个子节点：库备份、表备份、表空间备份和归档备份。各备份节点的右键菜单中提供了所有可操作的选项，包括新建备份、指定工作目录、备份恢复、备份校验、属性等。备份文件夹节点右键菜单如下：

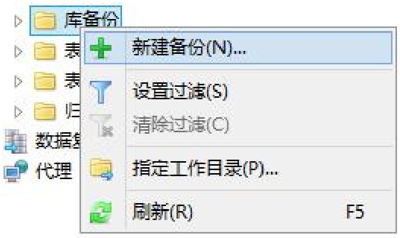


图 3.2 备份节点操作选项

备份节点右键菜单如下：

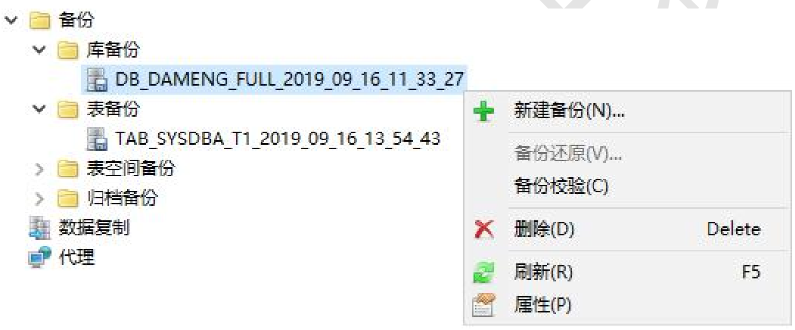


图 3.3 备份集节点操作选项

3.4.1.1 数据备份

数据备份包括库备份、表备份、表空间备份和归档备份，分别对应备份文件夹节点中库备份、表备份、表空间备份和归档备份的子节点。点开各子节点，显示当前已经存在的相应备份集名称。四者的备份流程基本一致，只是具体可配置的参数有些不同，下面以库备份为例介绍。

库备份节点右键菜单->新建备份，可以打开新建库备份对话框，首页显示的是常规选项页面，如图 3.4 所示：

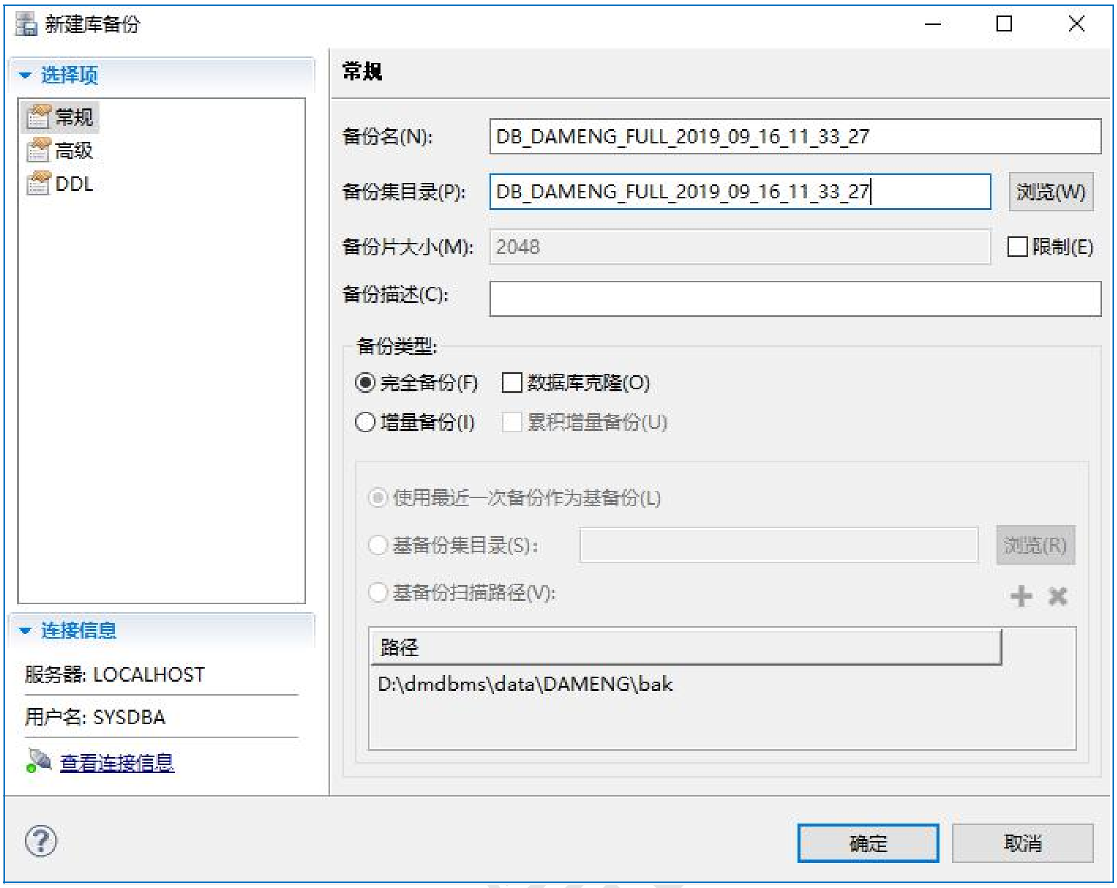


图 3.4 数据库备份常规选项页面

常规页面可配置库备份的常规参数信息，具体如下：

备份名：备份的名称。

备份集目录：备份集的目录，默认为系统默认备份集目录，即 DM 数据库安装目录下数据文件目录加备份集名称。

备份片最大大小：备份片的最大大小，如果备份文件超过该大小，则系统自动将备份文件分为多个文件。以兆为单位。不限制时，系统默认备份文件的最大大小为 2048M 。限制时，最小 128M，32 位系统最大 2G，64 位系统最大 128G。

备份描述：对备份文件的描述，用户可以根据需要填写。

备份类型：

- ✓ 完全备份：对数据库的完全备份，不依赖于其他备份，可以独立地进行数据库恢复。数据库克隆选项，表示进行数据库备份时，仅拷贝所有的元数据不拷贝数据。如对于数据库中的表来说，只备份表的定义不备份表中数据。
- ✓ 增量备份：在完全备份或增量备份的基础上进行的备份。进行增量备份时，允许用户自己指定基备份来进行备份，默认使用最近一次备份作为基备份，在该数据库不

存在任何备份时，备份类型不能为增量备份。累积增量备份选项的备份类型只能是基备份，缺省为差异增量备份。

库备份和表空间备份都可指定备份类型为增量备份，如果选择增量备份，可以为增量备份配置以下参数：

基备份目录：可以指定基备份集目录，默认系统会自己在指定的基备份扫描路径下查找最近一次备份作为基备份。

基备份扫描路径：指定基备份的扫描路径，可以同时指定多个，增量备份时必须保证依赖的所有基备份都可以在指定的基备份扫描路径中找到。

高级选项页面中可以配置一些其他的高级选项，如下图所示：

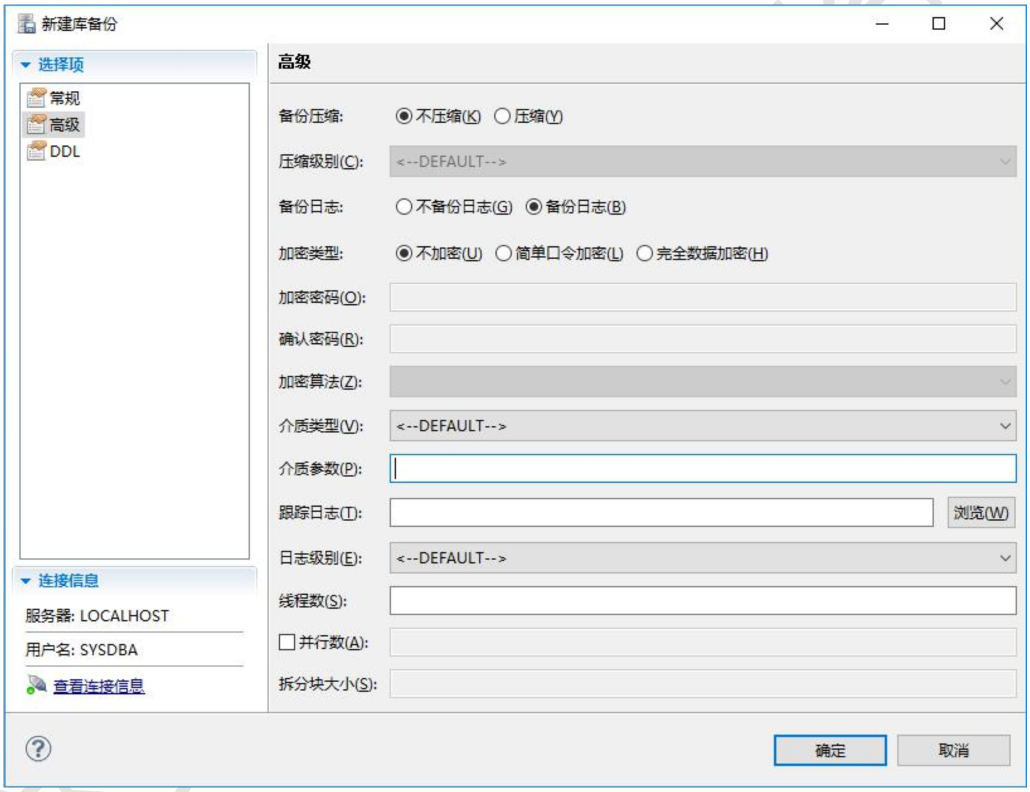


图 3.5 数据库备份高级选项页面

可为备份配置以下高级参数：

备份压缩：可以选择压缩或不压缩，通过压缩可以减少备份文件所需要的磁盘空间。默认为不压缩。

压缩级别：压缩的级别。可以设置压缩级别 0~9，1 表示 1 级压缩，9 表示 9 级压缩。压缩级别越高，压缩越慢，但压缩比越高。

备份日志：是否需要备份归档日志中的内容。默认备份日志。

加密类型：加密的类型，默认为不加密。

- ✓ 不加密：不对备份文件进行加密处理。
- ✓ 简单口令加密：对备份文件设置口令，但文件内容仍以明文存储。
- ✓ 完全数据加密：对备份文件进行完全的加密，备份文件以密文方式存储。

加密密码：当选择加密时，需要输入加密密码。

确认密码：当选择加密时，需要确认输入的加密密码。

加密算法：加密时使用的算法。在选择简单口令加密和完全数据加密的情况下适用。支持的加密算法参考数据库备份章节的参数说明。

介质类型：指定存储备份集的设备类型，默认选择磁盘，目前暂支持磁盘和磁带。

介质参数：使用第三方介质类型（TAPE 类型）时使用的参数。

跟踪日志：备份过程的跟踪日志路径。系统有默认路径，当日志级别为记录跟踪日志时启用。

日志级别：是否记录跟踪日志。默认不记录。

线程数：备份过程中数据处理过程线程的个数，取值范围 0~64，默认为 4。若指定为 0，则调整为 1；若指定超过当前系统主机核数，则调整为主机核数。线程数（TASK THREAD）*并行数（PARALLEL）不得超过 512。

并行数：指定是否执行并行备份，以及执行并行备份的并行数。并行数没有上限，但实际可使用的最大并行数就是目标备份数据文件的个数。若勾选但没指定并行数，则默认并行数为 4。若未勾选，则认为非并行备份。

拆分块大小用户指定对于大数据量数据文件并行备份时拆分块的大小，默认为 1GB，最小不能低于 512MB，当指定的<拆分块大小>小于 512MB 时，系统会自动调整为 512MB。若指定并行备份，但未指定<拆分块大小>，则直接使用默认拆分块大小进行拆分。当数据文件的大小小于<拆分块大小>时，不执行拆分；当数据文件的大小大于<拆分块大小>时，执行拆分。并行数不能大于拆分之后的总块数。

DDL 页面可以查看新建库备份的 SQL 语句，右下角的保存按钮也可以保存 SQL 脚本到文件中，如下图所示：

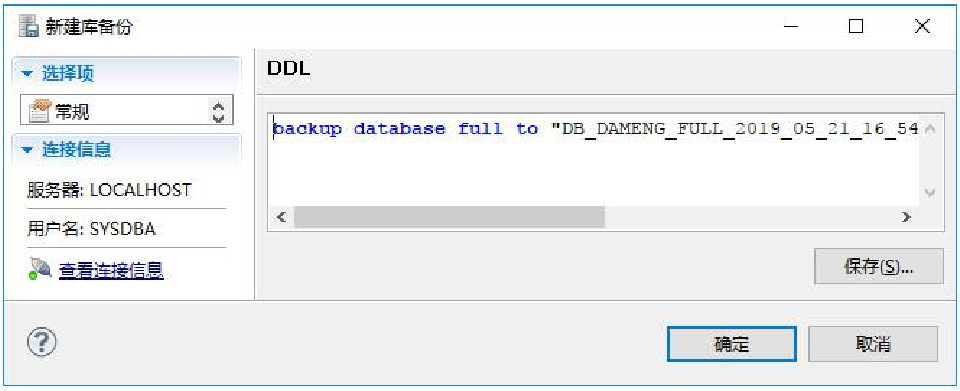


图 3.6 数据库备份查看 DDL 页面

3.4.1.2 备份管理

备份管理包括备份集信息、备份校验、备份删除和指定工作目录，其中备份删除是备份管理的主要功能。下面将分别对这些功能的使用进行介绍。

备份集查看

点击某个备份节点右键菜单->属性，可以打开备份属性查看对话框，属性列表中可以查看该备份集的信息，包括备份集的元数据信息集数据库信息，分别如图 3.7、图 3.8、图 3.9 所示：

达梦数据库信创大赛资料

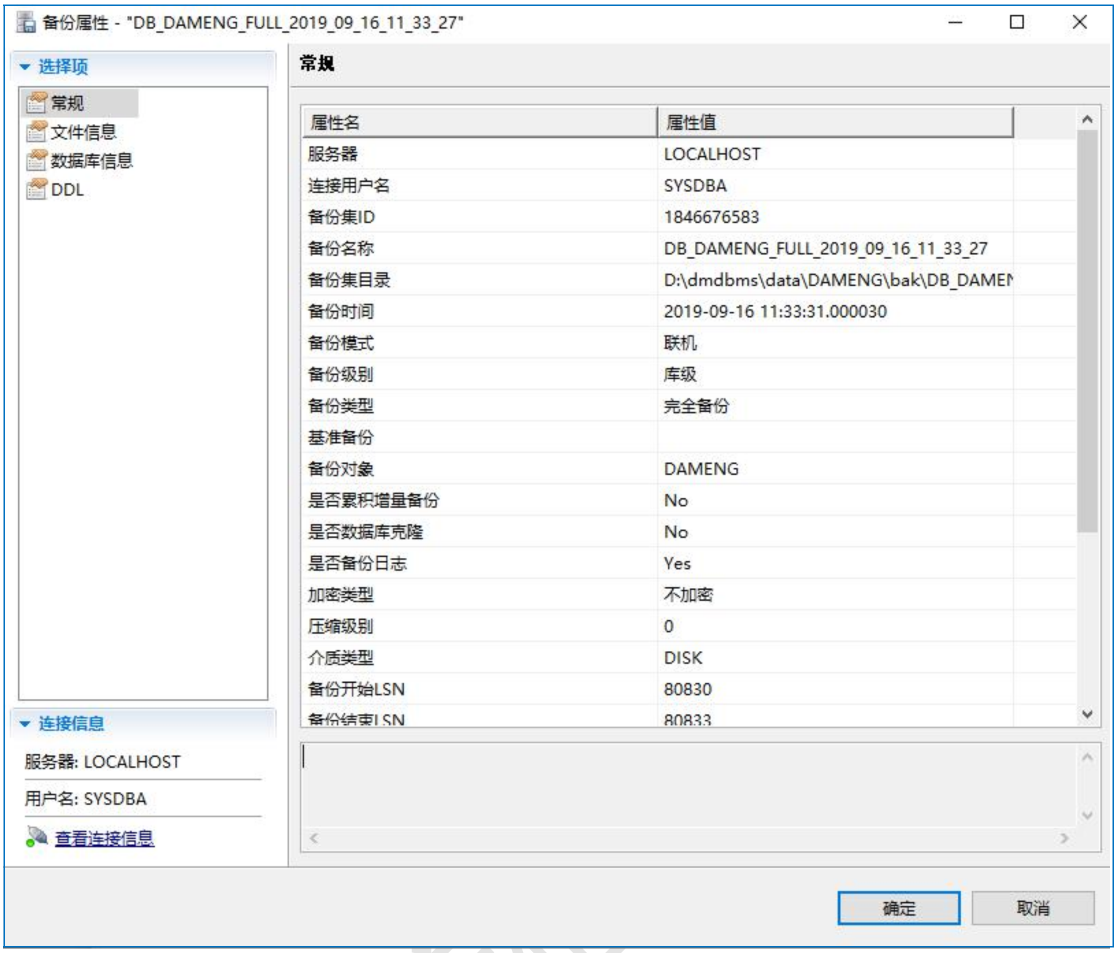


图 3.7 备份集常规信息显示页面

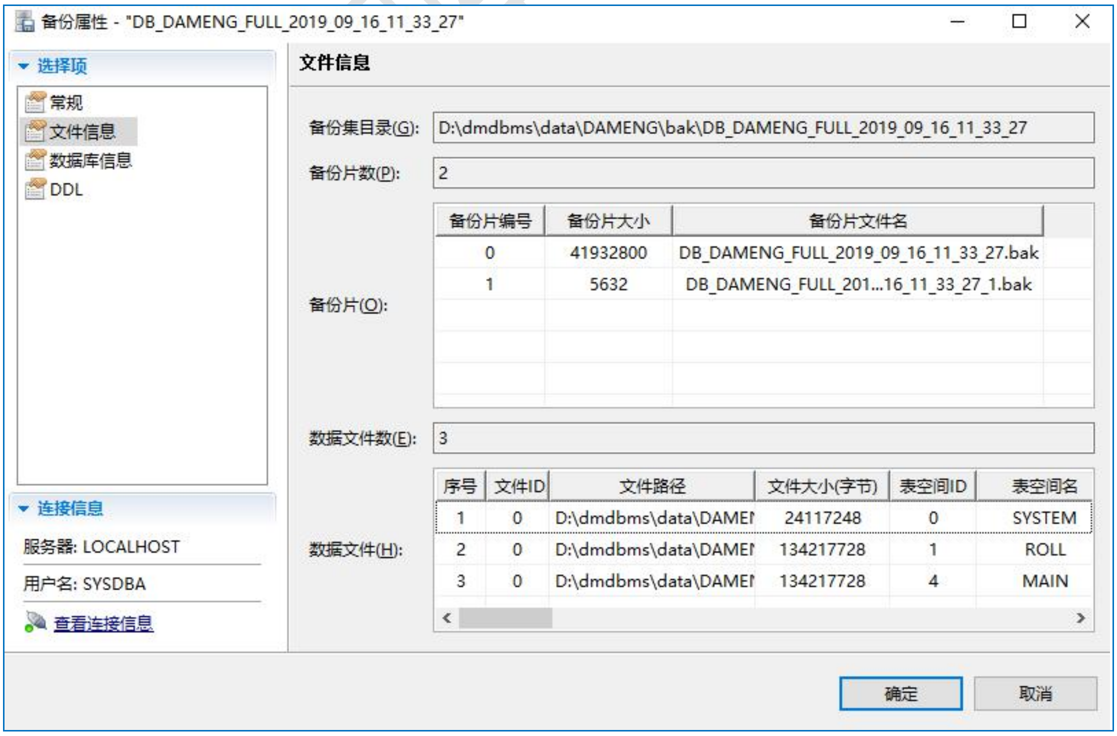


图 3.8 备份集文件信息界面

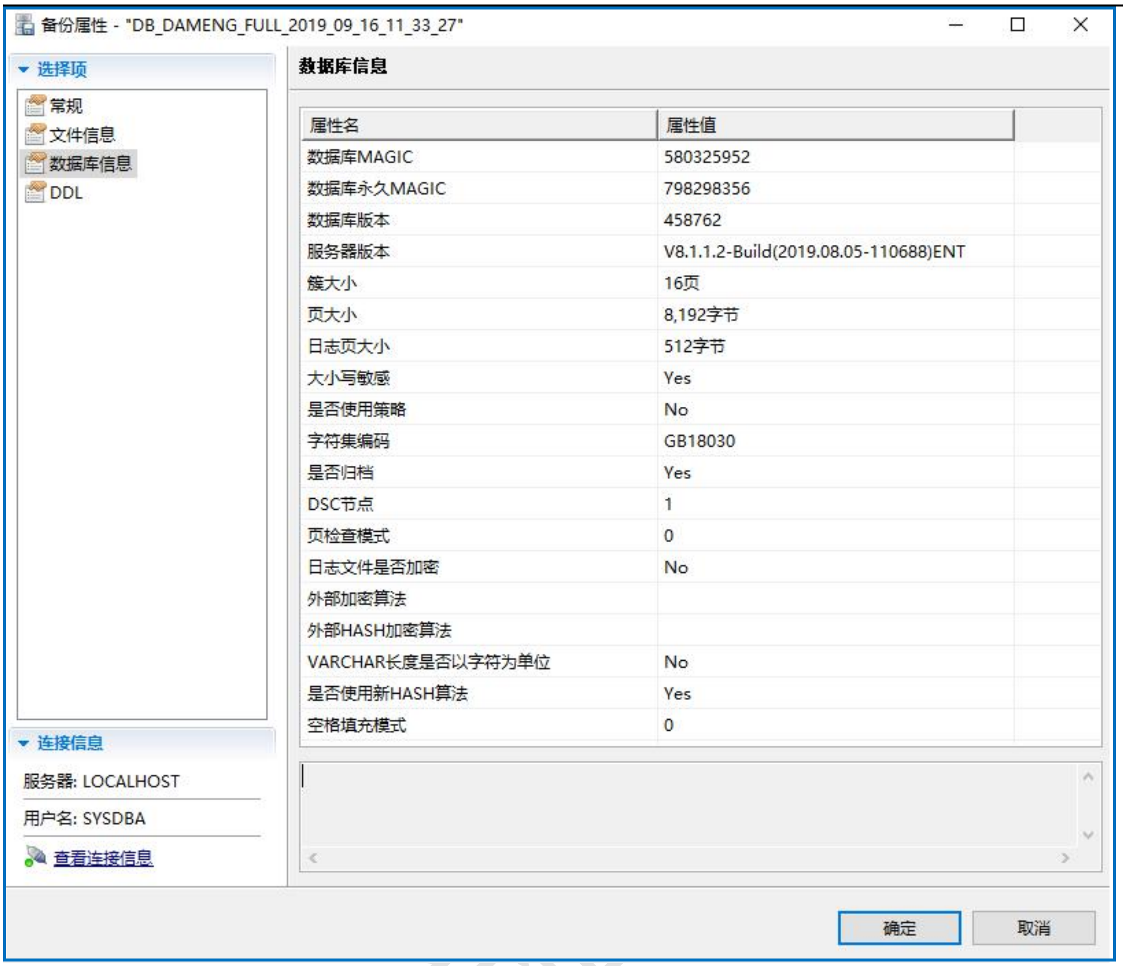


图 3.9 备份集数据库信息显示页面

备份校验

点击某备份节点右键菜单->备份校验，即可校验备份集的合法性。

备份删除

点击某备份节点右键菜单->删除，可以删除该备份集。也可以同时选中多个备份节点进行批量删除。

指定工作目录

点击备份文件夹节点右键菜单->指定工作目录，可以指定备份的工作目录，允许同时指定多个。如果设置了备份工作目录，备份文件夹下会显示所有工作目录下的所有备份。指定工作目录的对话框如下：

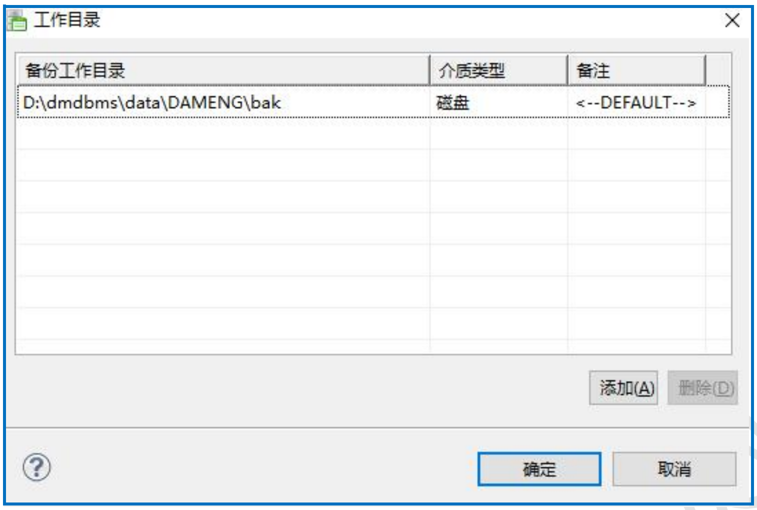


图 3.10 指定工作目录页面

点击添加按钮可以添加一个工作目录，选中一个工作目录点击删除按钮可以删除一个工作目录。系统有一个默认工作目录，该目录会一直保留不被删除。

3.4.1.3 数据还原

库备份和表空间不支持联机还原，只有表备份支持联机还原。表还原过程中表空间中其他的表还可以正常操作。

表备份节点右键菜单->备份还原，可以打开表空间备份还原对话框的常规页面如下图所示：

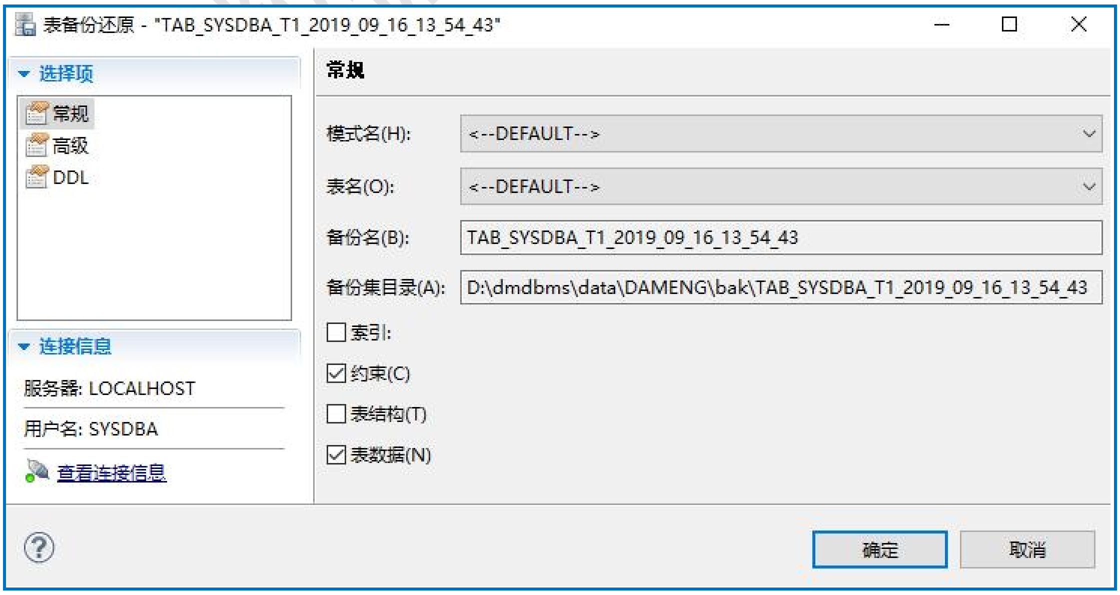


图 3.11 表还原常规选项页面

数据库信创大赛资料

可为表还原配置以下常规参数：

模式名：模式的名称。

表名：表的名称。

备份名：备份的名称，无需填写，系统自动获取。

备份集目录：备份集所在的目录，无需填写，系统自动获取。

WITH INDEX/WITHOUT INDEX：指定还原数据后是否重建二级索引，默认重建。

WITH CONSTRAINT/WITHOUT CONSTRAINT：指定还原数据后是否重建约束，默认重建。

索引：指定还原数据后是否重建二级索引。

约束：指定还原数据后是否重建约束。

表结构：指定只执行表结构还原，不还原数据。若未指定，则必须勾选表数据。

表数据：指定是否只执行表中数据还原；表数据还原要求还原目标表结构与备份集中完全一致，否则报错，所以在表数据还原时，建议勾选表结构还原执行，减少报错。若未指定表数据，则必须勾选表结构。表结构和表数据二者，至少要选一个。

高级选项页面如下：

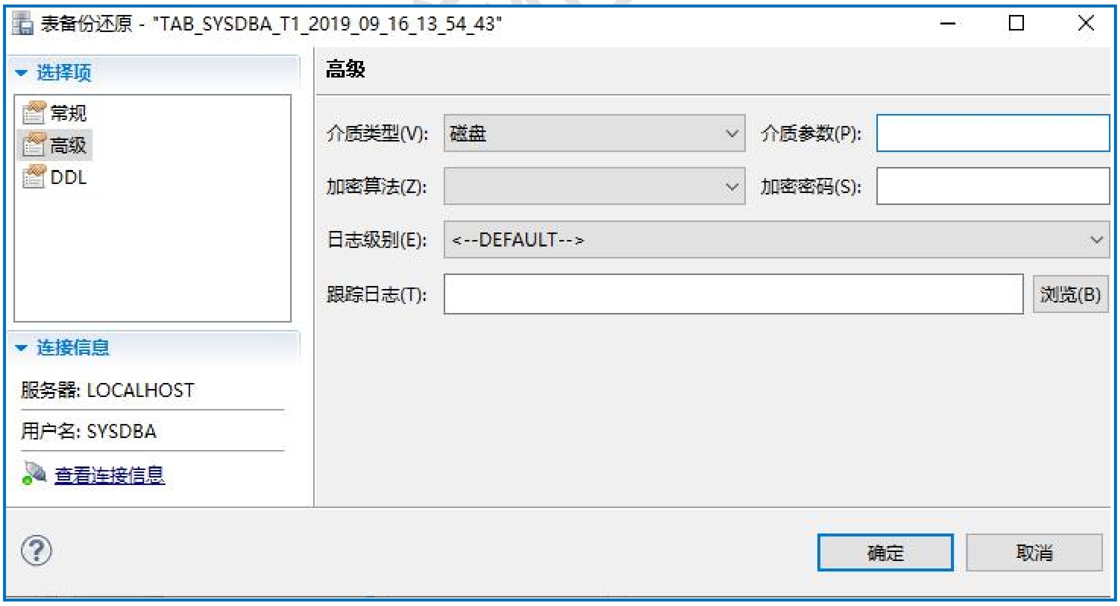


图 3.12 表还原常规选项页面

可配置以下参数：

介质类型：指存储备份集的介质类型，暂支持 DISK 和 TAPE，默认 DISK。

介质参数：使用第三方介质类型（TAPE 类型）时使用的参数。只对介质类型为 TAPE

信创大赛资料

时有效。

加密算法/加密密码： 如果备份时使用了加密，则还原时必须输入加密算法和加密密码。

日志级别：是否记录跟踪日志。有效值 1、2，默认为 1 表示不启用 trace，为 2 表示启用 trace。

跟踪日志：指定生成的 trace 文件名。

3.4.2 使用 CONSOLE 工具进行脱机备份还原

下面将介绍如何使用 DM 控制台工具 CONSOLE 来执行库级脱机备份与还原的操作。控制台主界面如下图所示。

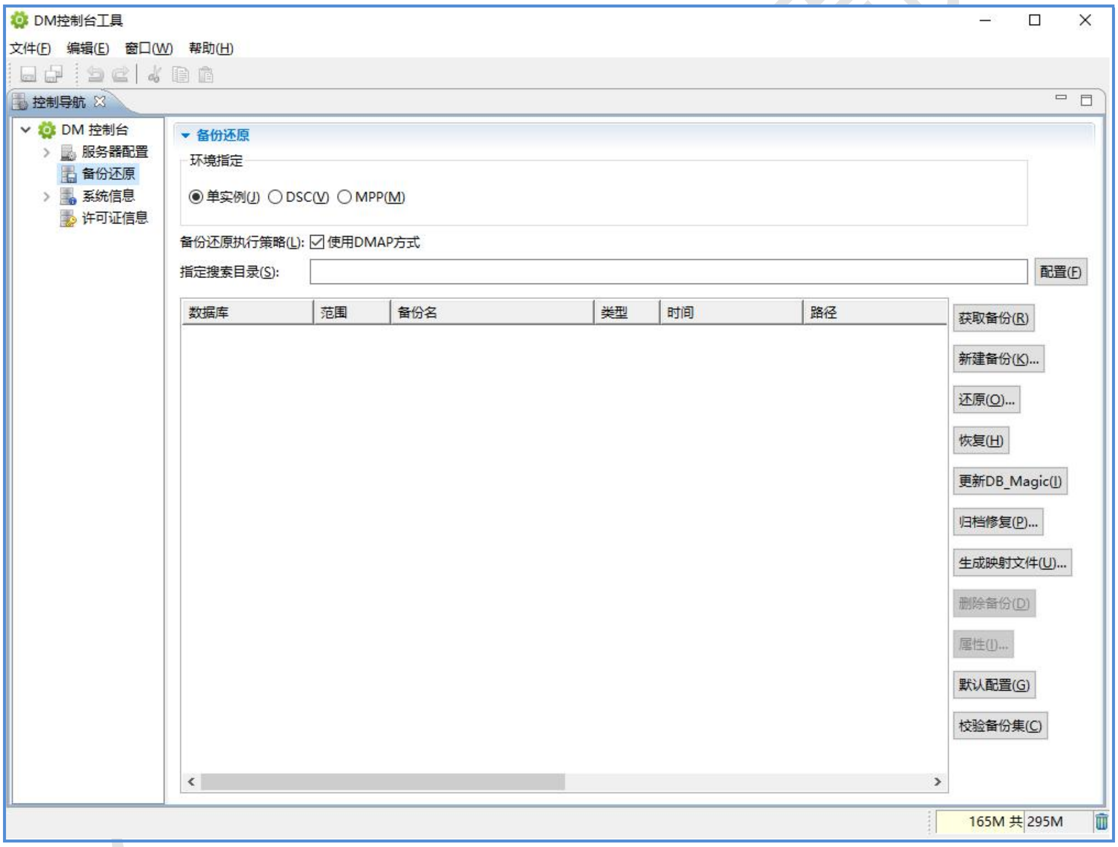


图 3.13 CONSOLE 工具备份还原主页面

3.4.2.1 数据备份

CONSOLE 工具的控制导航树中提供备份还原节点，单击备份还原节点可以打开备份还原管理界面。在备份还原管理界面中，点击新建备份按钮。打开新建备份对话框，如下图所示。

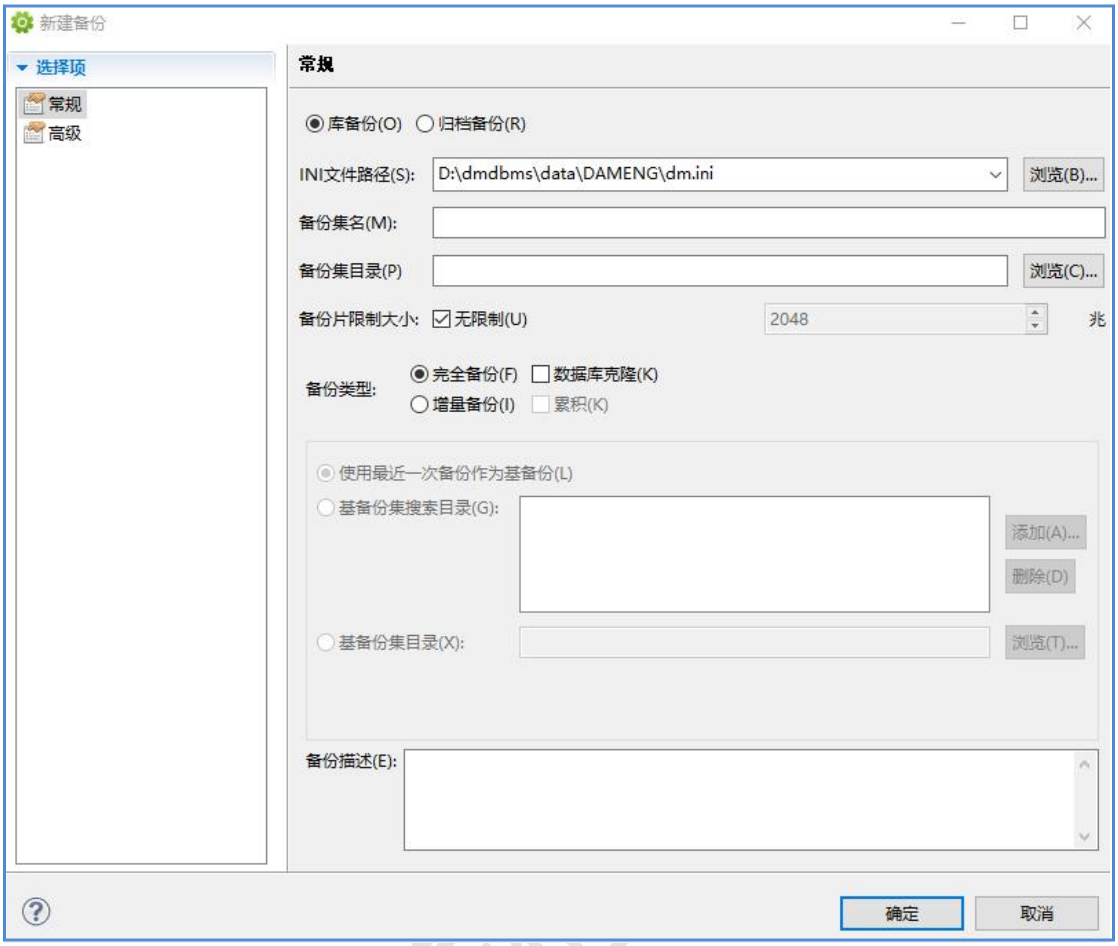


图 3.14 数据库备份常规选项页面-库备份

需要为脱机库备份配置以下参数：

INI 文件路径：必填，待备份目标数据库 dm.ini 文件路径。

备份集名：选填，指定生成备份集名称，若未指定，则使用约定方法随机生成。

备份集目录：选填，指定当前备份集生成目录，若未指定，则在默认备份目录中生成备份集目录。

备份片限制大小：默认无限制，取消选中无限制，则可以设置备份片的限制大小，以兆为单位，最小 128M，32 位系统最大 2G，64 位系统最大 128G。

备份类型：完全备份和增量备份。数据库克隆是完全备份的一个选项，表示进行数据库备份时，仅拷贝所有的元数据不拷贝数据。默认选完全备份，在已经存在完全备份的情况下，可以选择增量备份。

基备份集搜索目录：选填，并且仅适用于增量备份。

基备份集目录：选填，并且仅适用于增量备份。

备份描述：选填，根据实际情况对备份集进行描述。

根据实际需要配置完上述参数后，点击确定按钮就开始执行脱机备份操作了。备份操作可能需要花费一些时间，备份结束后，如果备份成功则弹出备份成功的对话框；如果备份失败，则弹出对话框提示失败原因。

3.4.2.2 备份管理

查看备份集

在备份还原管理页面中，指定搜索目录，点击获取备份按钮，即可获取备份集列表。操作步骤如图 3.16、图 3.17、图 3.18 所示。

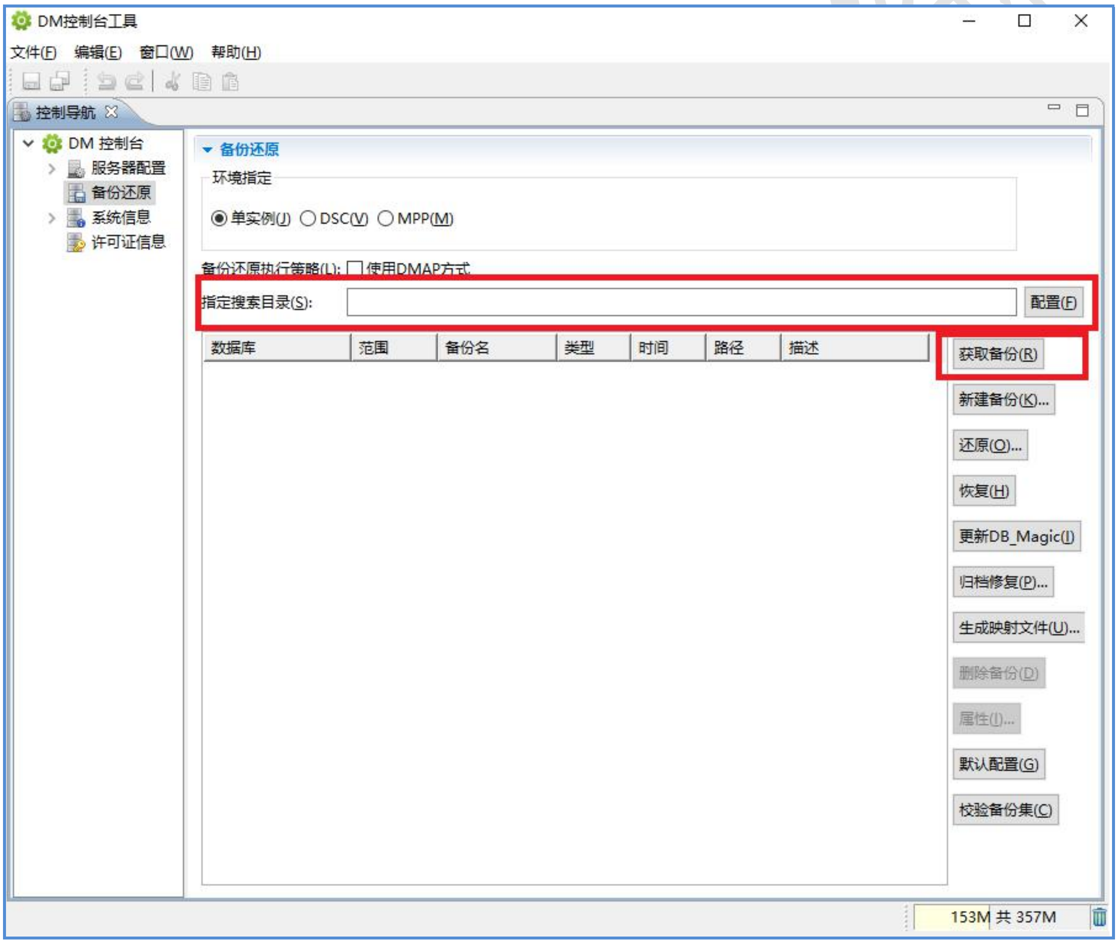


图 3.17 指定搜索目录并获取备份

可以获取到指定搜索目录下的所有备份集，显示在备份列表中。选择要查看的备份集，点击属性按钮。

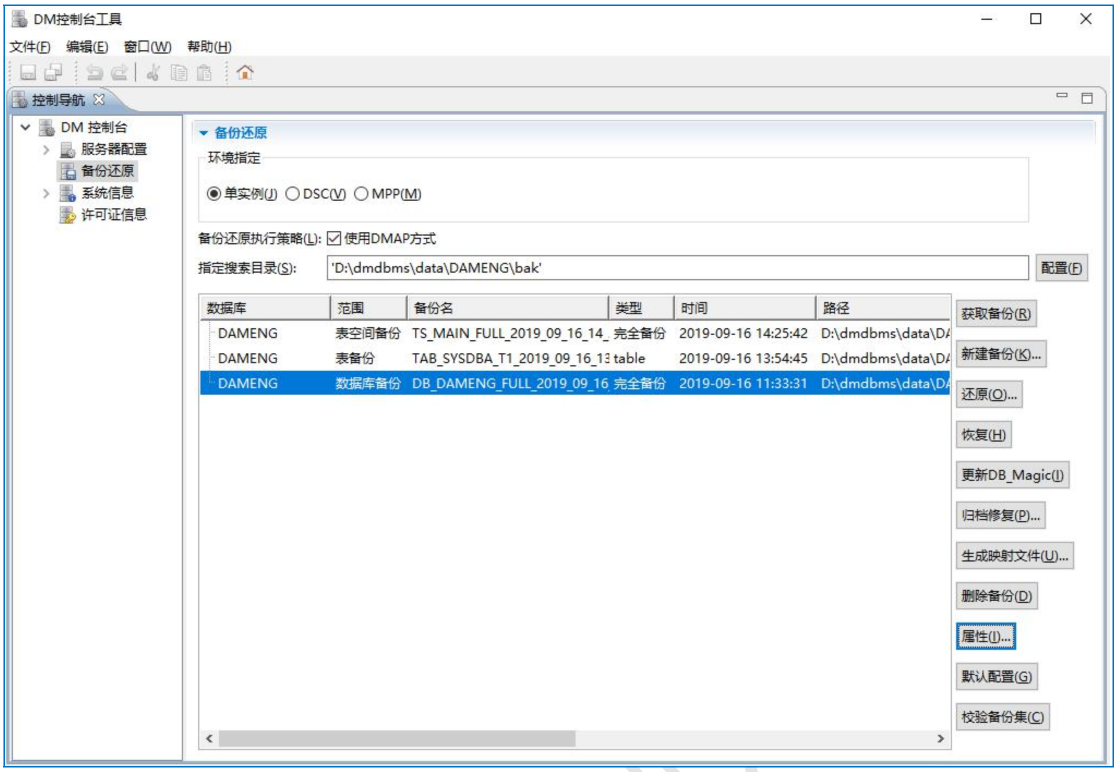


图 3.18 备份集列表显示页面

打开备份属性对话框，可查看备份属性。备份属性分为常规、元信息、DSC 信息、文件信息、数据库信息几类。其中 DSC 信息只有 DSC 环境的备份集才有。常规属性主要显示备份集相关的属性，如图 3.19 所示。

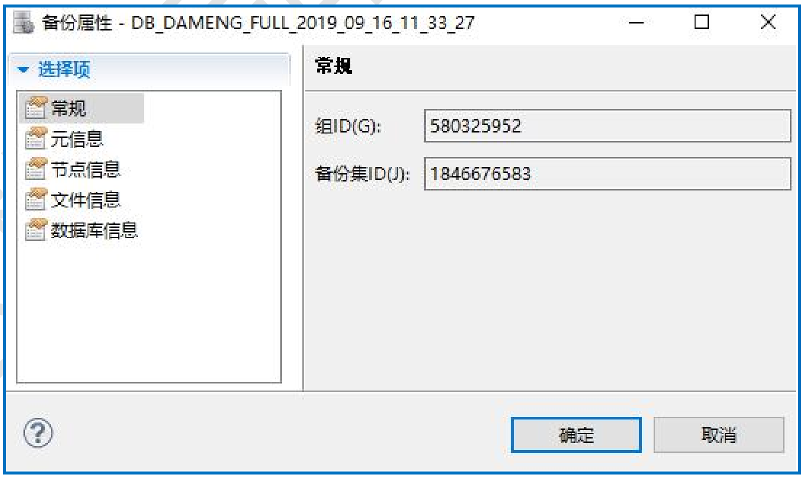
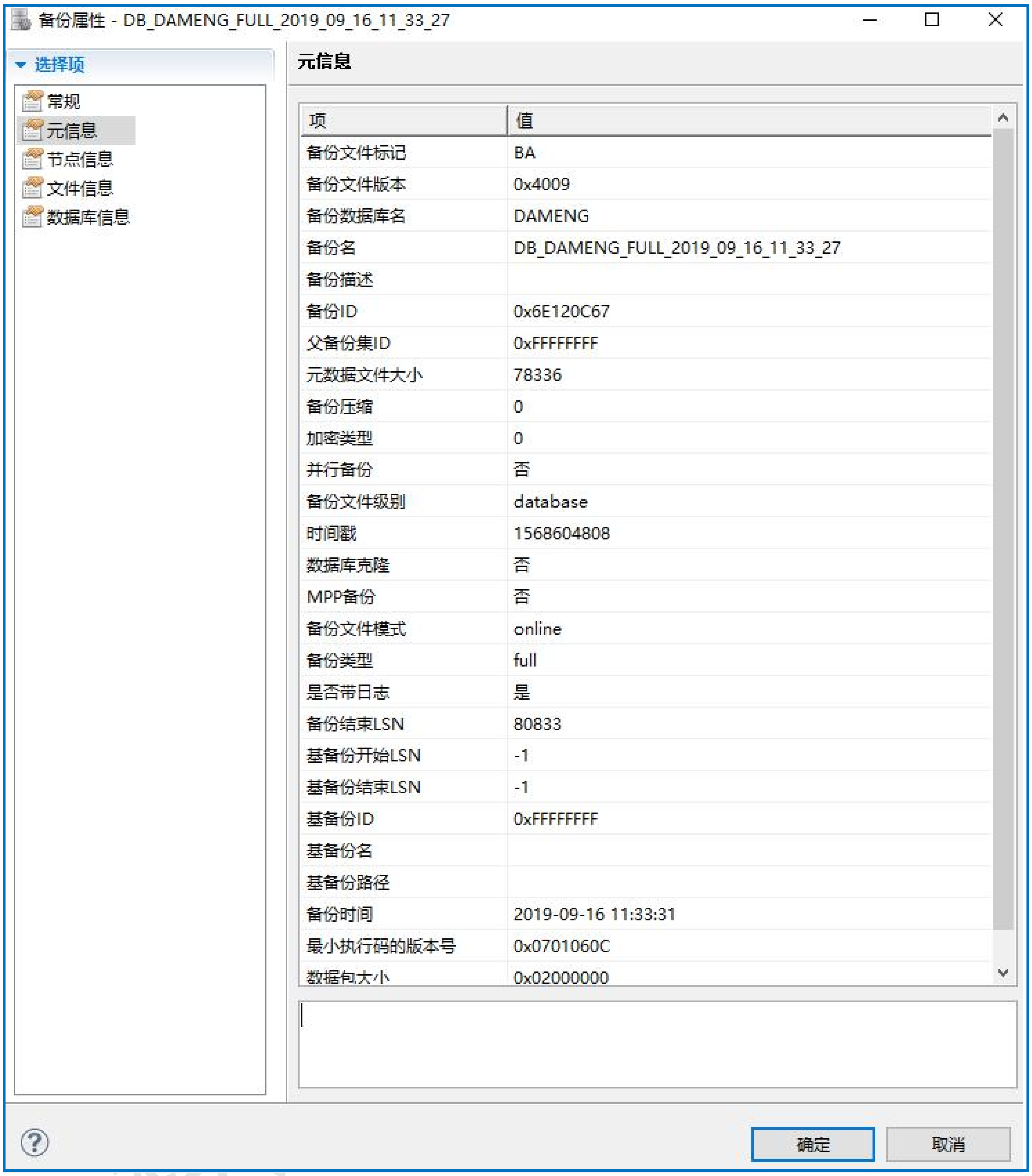


图 3.19 备份集-常规信息

元信息主要显示备份集本身相关的信息如备份是否联机备份、备份的范围、备份的加密信息，以及备份的压缩信息等，如下图所示。



项	值
备份文件标记	BA
备份文件版本	0x4009
备份数据库名	DAMENG
备份名	DB_DAMENG_FULL_2019_09_16_11_33_27
备份描述	
备份ID	0x6E120C67
父备份集ID	0xFFFFFFFF
元数据文件大小	78336
备份压缩	0
加密类型	0
并行备份	否
备份文件级别	database
时间戳	1568604808
数据库克隆	否
MPP备份	否
备份文件模式	online
备份类型	full
是否带日志	是
备份结束LSN	80833
基备份开始LSN	-1
基备份结束LSN	-1
基备份ID	0xFFFFFFFF
基备份名	
基备份路径	
备份时间	2019-09-16 11:33:31
最小执行码的版本号	0x0701060C
数据包大小	0x02000000

图 3.20 备份集-元数据信息

数据库信息显示备份库的固有信息。如下图所示：

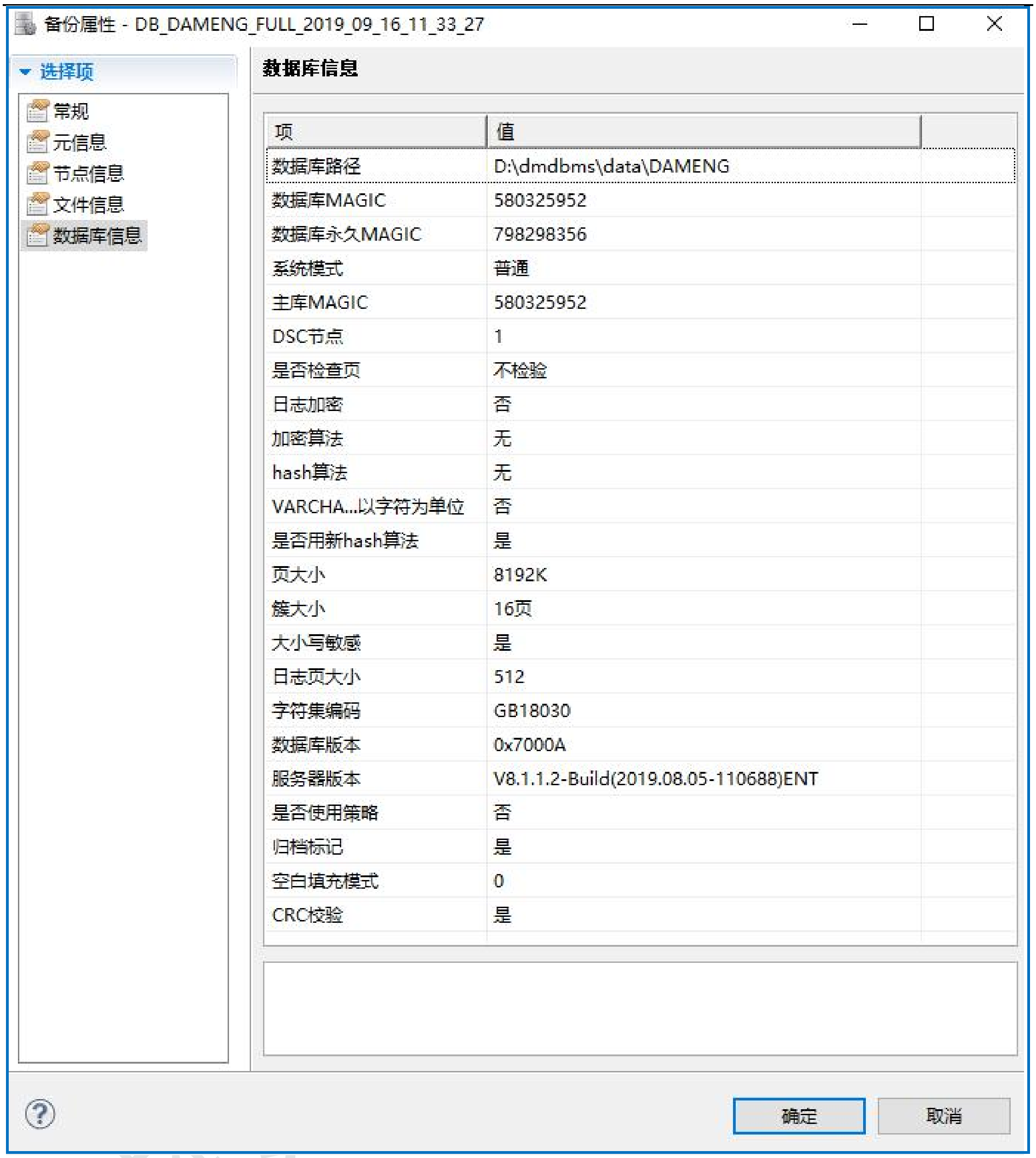


图 3.23 备份集-数据库信息

3.4.2.3 数据还原

数据库还原

在备份还原管理主界面中，点击还原按钮，打开备份还原对话框。还原有三种类型：库还原（图 3.23）、归档还原（图 3.24）、表空间还原（图 3.25）和直接还原（图 3.26）。

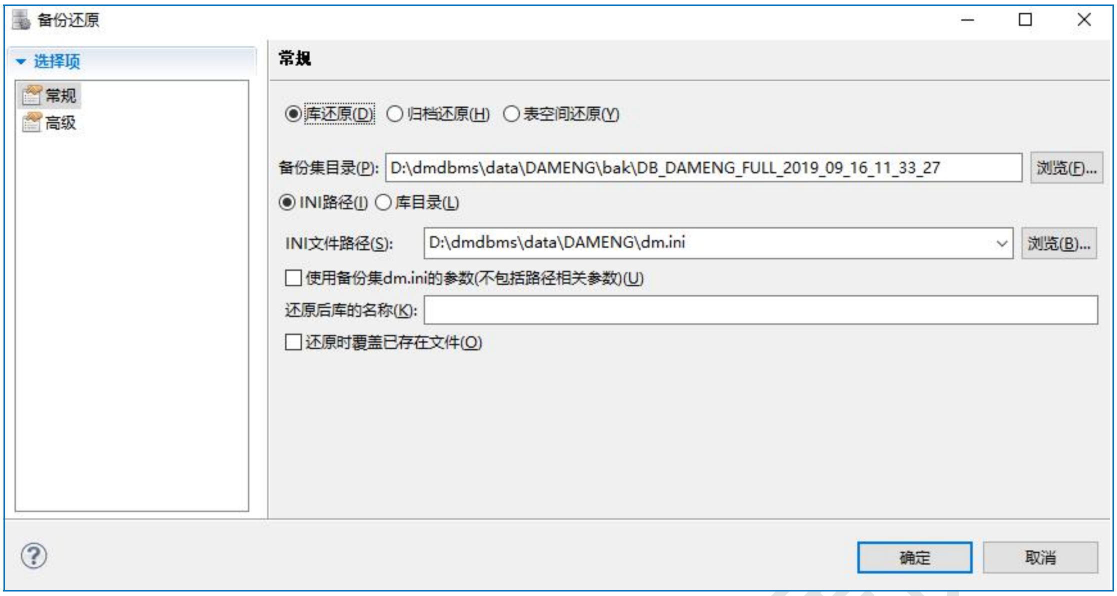


图 3.24 备份集还原常规页面--库还原

需要配置以下参数：

备份集目录：指定用于还原目标数据库的备份目录。若指定为相对路径，会在基备份集搜索目录下搜索备份集。

INI 文件路径：指定还原库目标的 dm.ini 文件路径。此时，还可以使用备份集 dm.ini 的参数，表示会将备份集中备份的 dm.ini 参数（除路径相关的 INI 参数外），拷贝到当前 dm.ini 上，取而代之。

库目录：待还原的库所在目录，该路径必须已存在。此时，还可以选用还原时覆盖已存在的文件，表示如果待还原的文件已存在，直接覆盖，否则报错。

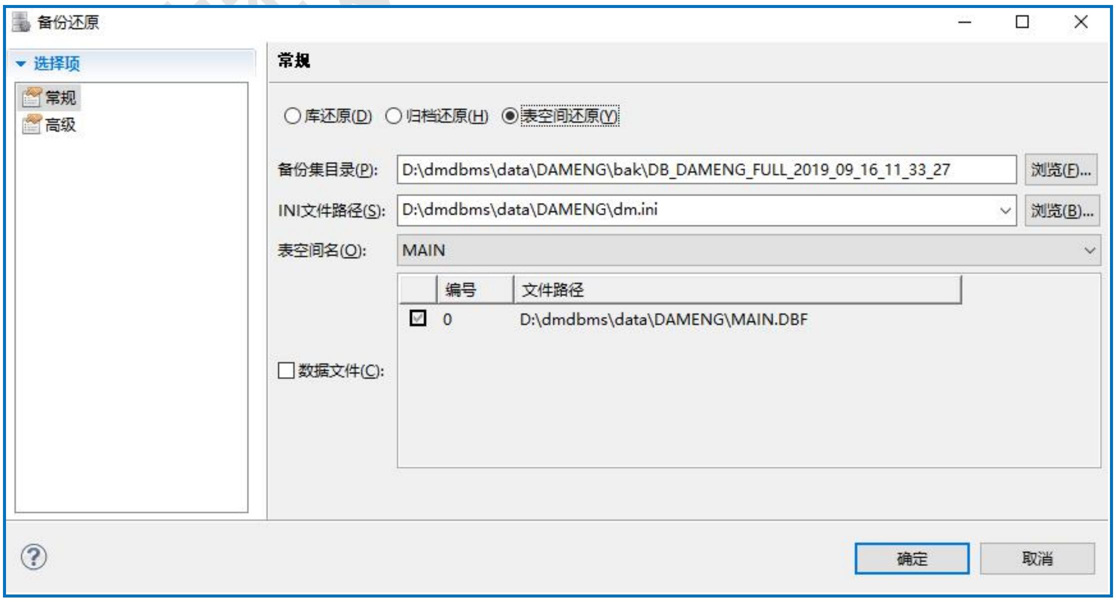


图 3.26 备份集还原常规页面--表空间还原

需要配置以下参数：

备份集目录：指定用于还原目标数据库的备份集目录。若指定为相对路径，会在基备份集搜索目录下搜索备份集。

INI 文件路径：指定还原库目标的 dm.ini 文件路径。

表空间名:指定还原的表空间,可以是用户表空间,也可以是 SYSTEM 系统表空间和 ROLL 回滚表空间。

数据文件：还原指定的数据文件。可以指定数据文件编号或数据文件路径。文件编号，对应动态视图 V$DATAFILE 中 ID 列的值；文件路径，对应动态视图 V$DATAFILE 中 PATH 或者 MIRROR_PATH 列的值，也可以仅指定数据文件名称（相对路径），与表空间中数据文件匹配时，会使用 SYSTEM 目录补齐。

归档日志目录：本地归档日志搜索目录，若未指定，则仅使用目标库配置的第一个本地归档。

截止时间点：恢复数据库到指定的时间点。

截止 LSN：恢复数据库到指定的 LSN。

3.4.2.4 数据恢复

在备份还原管理界面中，点击恢复按钮，打开备份恢复对话框。有两种级别的恢复：库恢复和表空间恢复。

3.4.2.4.1 库恢复

数据库恢复有两种类型。

- 从备份集恢复
- 指定归档恢复

以下分别进行介绍。

从备份集恢复

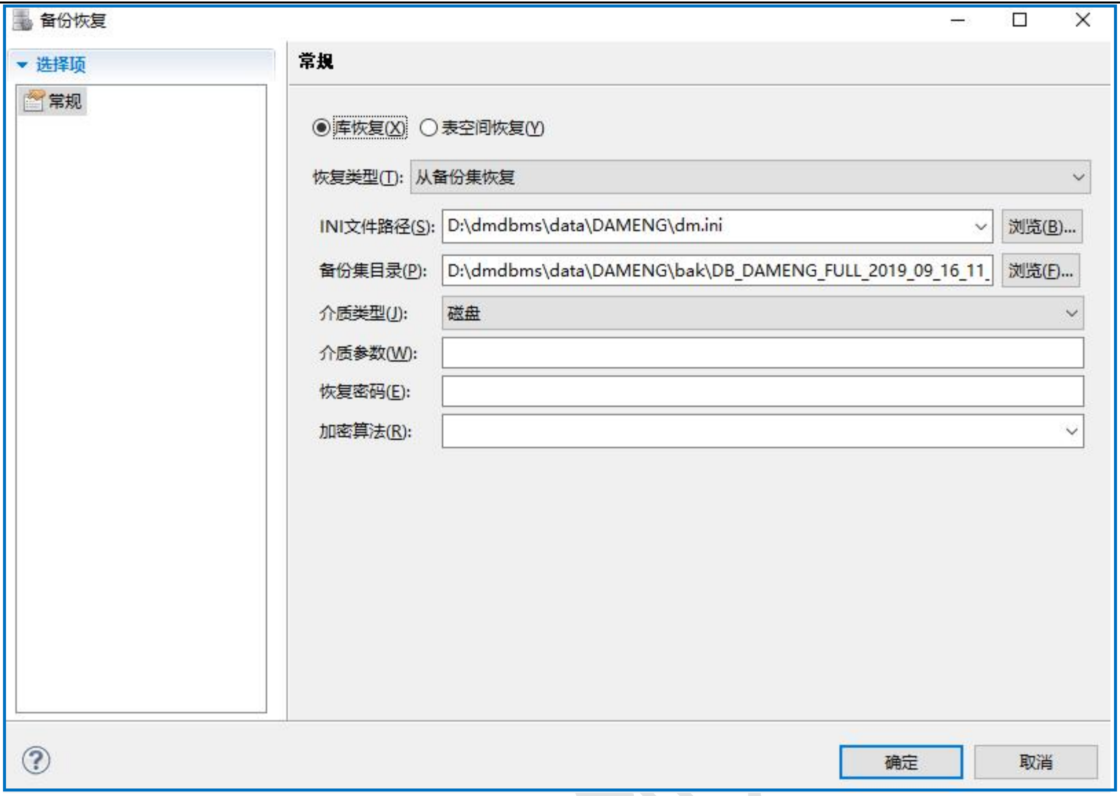


图 3.29 从备份集恢复页面

如果选择从备份集恢复，则需要配置以下参数：

INI 文件路径：必填，待恢复目标数据库 dm.ini 文件路径。

备份集目录：必填，指定待恢复备份集目录。

介质类型：指存储备份集的介质类型，暂支持 DISK 和 TAPE，默认 DISK。

指定归档恢复

指定归档恢复，主要有两种类型：

1）没有经过还原操作，直接在目标库上执行恢复操作。这种恢复主要针对一些已经滞后的库，用最新的归档把目标库恢复到某个时间点，或者恢复到最新的状态。

2）经过了还原操作，这时需要借助归档把目标恢复到一个指定的时间点。可能还原的备份集是 WITHOUT LOG 的备份集，没有办法执行从备份集恢复；或者希望恢复到更新的状态，而从备份集恢复又不能够满足要求，所以需要借助归档恢复到某个时间点。

在未指定恢复时间或者恢复 LSN 号时，指定归档恢复默认恢复到最新状态，有多少归档就会重做多少。

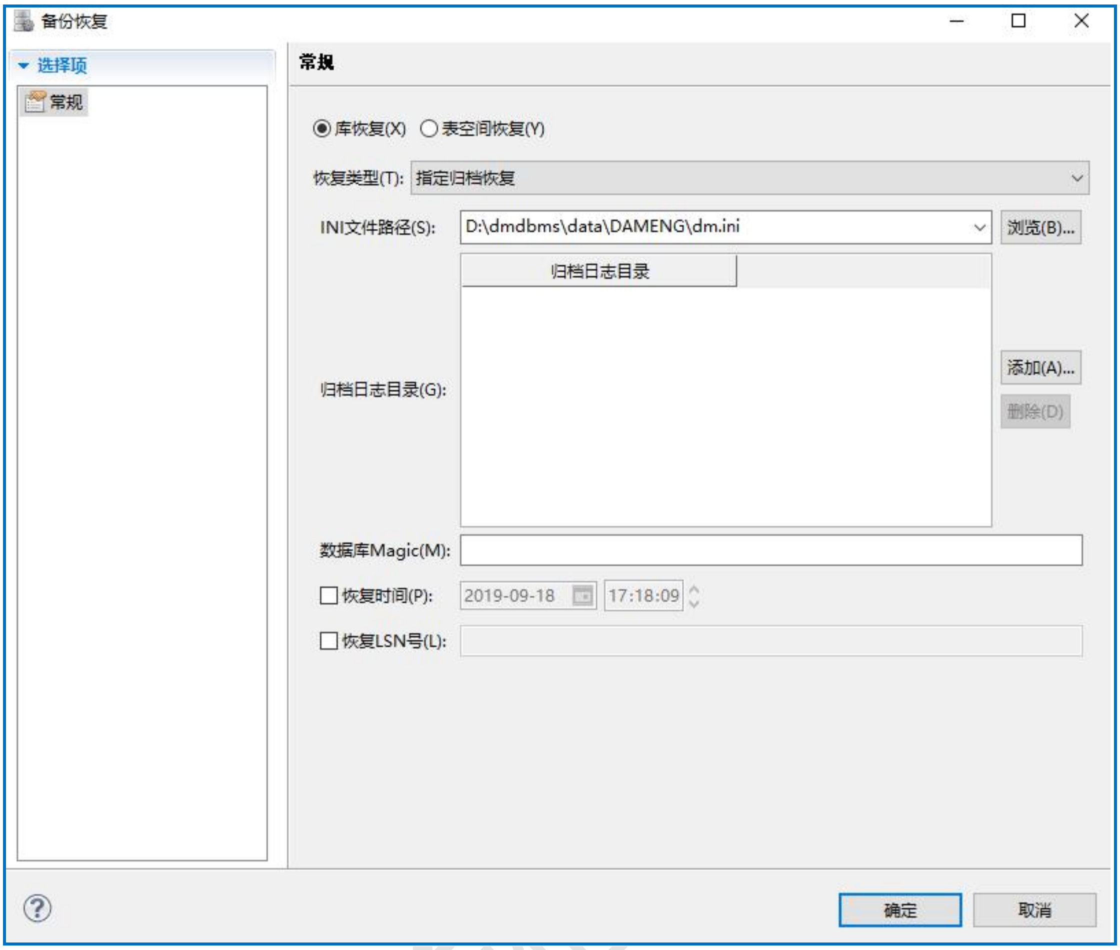


图 3.30 指定归档恢复页面

如果选择指定归档恢复，则需要配置以下参数：

INI 文件路径：必填，待恢复目标数据库 dm.ini 文件路径。

归档日志目录：必填，通过添加、删除按钮配置归档日志目录。

数据库 Magic：指定本地归档日志对应数据库的 Magic，若不指定，则默认使用目标恢复数据库的 Magic。

恢复时间：指定重做本地归档结束时间。

恢复 LSN 号：指定重做本地归档的结束 LSN 值。

根据实际需要选择对应的恢复类型，配置对应的参数后，点击确定按钮即开始执行对应恢复操作。如果恢复成功，则弹出恢复成功的对话框；如果恢复失败，则弹出对话框提示失败原因。

3.4.2.5 数据库更新

更新 DB_MAGIC 是数据库还原之后，必须执行的一步。表空间等其他类型的还原不需要执行这一步。

对于联机备份的备份集，对目标库执行完还原操作后，如果仅希望把目标库恢复到备份的时间点，可以借助更新 DB_MAGIC 把目标库恢复到备份结束的时间点。这种情况的恢复要求联机备份集是使用 WITHOUT LOG 类型的备份，且要注意只能恢复到备份结束的时间点。

对于脱机备份的备份集，或者联机备份的备份集中不需要重做联机日志的情况，在执行完还原操作后允许执行更新 DB_MAGIC 操作，仅仅把目标库恢复到备份结束的时间点。



图 3.32 更新 DB_MAGIC 页面

如果选择更新 DB_MAGIC 恢复，则需要配置以下参数：

INI 文件路径：必填，待恢复目标数据库 dm.ini 文件路径。

4 逻辑导入导出功能简介

DM 数据库的备份还原包括两种类型：物理备份还原和逻辑备份还原。物理备份还原是对数据库的操作系统物理文件（如数据文件、控制文件和日志文件等）的备份还原。例如使用 RMAN 工具进行的备份还原。逻辑备份还原是对数据库逻辑组件（如表、视图和存储过程等数据库对象）的备份还原。例如使用 dexp 和 dimp 进行的备份还原。逻辑备份还原正是本书所要介绍的内容。

逻辑导出（dexp）和逻辑导入（dimp）是 DM 数据库的两个命令行工具，分别用来实现对 DM 数据库的逻辑备份和逻辑还原。逻辑备份和逻辑还原都是在联机方式下完成，联机方式是指数据库服务器正常运行过程中进行的备份和还原。dexp 和 dimp 是 DM 数据库自带的

工具，只要安装了 DM 数据库，就可以在安装目录/dmdbms/bin 中找到。

逻辑导出和逻辑导入数据库对象分为四种级别：数据库级、用户级、模式级和表级。四种级别独立互斥，不能同时存在。四种级别所提供的功能：

- 数据库级（FULL）：导出或导入整个数据库中的所有对象。
- 用户级（OWNER）：导出或导入一个或多个用户所拥有的所有对象。
- 模式级（SCHEMAS）：导出或导入一个或多个模式下的所有对象。
- 表级（TABLES）：导出或导入一个或多个指定的表或表分区。

达梦数据库信创大赛资料

5 dexp 逻辑导出

dexp 工具可以对本地或者远程数据库进行数据库级、用户级、模式级和表级的逻辑备份。备份的内容非常灵活，可以选择是否备份索引、数据行和权限，是否忽略各种约束（外键约束、非空约束、唯一约束等），在备份前还可以选择生成日志文件，记录备份的过程以供查看。

5.1 使用 dexp 工具

dexp 工具需要从命令行启动。在 cmd 命令行工具中找到 dexp 所在安装目录/dmdbms/bin，输入 dexp 和参数后回车。参数在下一节详细介绍。

语法如下：

```
dexp  PARAMETER=<value> { PARAMETER=<value> }
```

PARAMETER：dexp 参数。多个参数之间排列顺序无影响，参数之间使用空格间隔。

<value>：参数取值。

例如，将用户名和密码均为 SYSDBA，IP 地址为 192.168.0.248，端口号为 8888 的数据库采用 FULL 方式完全导出。/user/data 为数字证书路径。导出文件名为 db_str.dmp，导出的日志文件名为 db_str.log，导出文件的路径为/mnt/dexp/data。

```
./dexp USERID=SYSDBA/SYSDBA@192.168.0.248:8888#/user/data FILE=db_str.dmp
DIRECTORY=/mnt/dexp/data  LOG=db_str.log FULL=Y
```

5.2 dexp 参数一览表

本节提供 dexp 的参数一览表，供用户快速参考。

表 2.1 dexp 参数一览表

参数	含义	备注
USERID	用户名/口令@主库名:端口号#证书路径。例如： SYSDBA/SYSDBA	必选。其中主库名、端口号和证书路径为可

	*MPP_TYPE@server:5236#ssl_path@ssl_pwd	选项
FILE	明确指定导出文件名称	可选。如果缺省该参数，则导出文件名为 dexp.dmp
DIRECTORY	导出文件所在目录	可选
FULL	导出整个数据库(N)	可选，四者中选其一。缺省为 SCHEMAS
OWNER	用户名列表，导出一个或多个用户所拥有的所有对象	
SCHEMAS	模式列表，导出一个或多个模式下的所有对象	
TABLES	表名列表，导出一个或多个指定的表或表分区	
FUZZY_MATCH	TABLES 选项是否支持模糊匹配(N)	可选
QUERY	用于指定对导出表的数据进行过滤的条件。	可选
PARALLEL	用于指定导出的过程中所使用的线程数目	可选
TABLE_PARALLEL	用于指定导出每张表所使用的线程数,在 MPP 模式下会转换成单线程	可选
TABLE_POOL	用于设置导出过程中存储表的缓冲区个数	可选
EXCLUDE	1．导出内容中忽略指定的对象。对象有 CONSTRAINTS、INDEXES、ROWS、TRIGGERS 和 GRANTS。 比如：EXCLUDE= (CONSTRAINTS,INDEXES) 2．忽略指定的表，使用 TABLES:INFO 格式，如果使用表级导出方式导出，则使用 TABLES:INFO 格式的 EXCLUDE 无效。例如：EXCLUDE= TABLES: table1,table2 3．忽略指定的模式，使用 SCHEMAS:INFO 格式，如果使用表级,模式级导出方式导出,则使用 SCHEMAS:INFO 格式的 EXCLUDE 无效。例如：EXCLUDE=SCHEMAS: SCH1,SCH2	可选
INCLUDE	导出内容中包含指定的对象 例如：INCLUDE=(CONSTRAINTS,INDEXES)	可选

达梦数据库信创大赛资料

	或者 INCLUDE=TABLES:table1,table2	
CONSTRAINTS	导出约束 (Y)	可选。 此处单独设置与和 EXCLUDE/INCLUDE 中批量设置功能一样。设置一个即可
TABLESPACE	导出的对象定义是否包含表空间(N)	
GRANTS	导出权限 (Y)	
INDEXES	导出索引 (Y)	
TRIGGERS	导出触发器（Y）	
ROWS	导出数据行 (Y)	
LOG	明确指定日志文件名称	可选，如果缺省该参数，则导出文件名为 dexp.log
NOLOGFILE	不使用日志文件(N)	可选
NOLOG	屏幕上不显示日志信息(N)	可选
LOG_WRITE	日志信息实时写入文件 (N)	可选
DUMMY	交互信息处理：打印(P)，所有交互都按 YES 处理(Y),NO(N)。默认为 NO，不打印交互信息。	可选
PARFILE	参数文件名，如果 dexp 的参数很多，可以存成参数文件	可选
FEEDBACK	每 x 行显示进度 (0)	可选
COMPRESS	是否压缩导出数据文件(N)	可选
ENCRYPT	导出数据是否加密 (N)	可选， 和 ENCRYPT 同时使用
ENCRYPT_PASSWORD	导出数据的加密密钥	
ENCRYPT_NAME	导出数据的加密算法	可选。 和 ENCRYPT 、ENCRYPT_PASSWORD 同时使用。缺省为 RC4
FILESIZE	用于指定单个导出文件大小的上限。可以按字节[B]、K[B]、M[B]、G[B]的方式指定大小	可选
FILENUM	多文件导出时，一个模板可以生成文件数，范围为[1,99],默认 99	可选

达梦数据库信创大赛资料

DROP	导出后删除原表，但不级联删除 （N）	可选
DESCRIBE	导出数据文件的描述信息，记录在数据文件中	可选
LOCAL	MPP 环境下使用 MPP_LOCAL 方式登录(N)	可选
HELP	显示帮助信息	可选

注意：每个参数含义后面的括号内为(N)则表示该参数缺省为否，为(Y)则表示为是，虽然参数大部分为可选参数，但后台这些参数都会传到服务器，此时参数的值就是缺省值。

5.3 dexp 参数详解

本节详细介绍 dexp 的主要参数。

5.3.1 特殊参数

特殊参数中介绍了必选参数的和最常用的可选参数。

USERID

USERID 用于连接服务器。必选参数。

语法如下：

```
USER/PWD@SERVER:PORT
```

USER/PWD：用户名和密码，用户名必写，密码缺省为 SYSDBA。

SERVER:PORT：服务器 IP 地址和端口号。缺省情况下默认为本地服务器和端口号 LOCALHOST:5236。当服务器为本机时，SERVER:PORT 可直接写 LOCALHOST。例如：192.168.0.248:8888。

当连接其他服务器时，SERVER:PORT 需写上 IP 地址和 PORTNUM。

达梦数据库信创大赛资料

小窍门： **PWD 密码中带有特殊字符（/,@.:等），需要外加三层双引号（""""""）进行转义。**

PWD 密码中带有特殊字符的具体转义处理规则：

登录时，密码中含有/,@.:等特殊字符时，密码需要外加三层双引号（""""""）进行转义。下面使用一个例子进行解析。例如：密码为 aaa/aaa，首先，dexp 要求对含有特殊字符的密码内部进行转义（外加一层双引号），写成"aaa/aaa"。其次，要将转义后的内容作为一个整体传给 dexp（外加一层双引号），写成""aaa/aaa""。最后，整个内容需要作为一个整体传给操作系统（外加一层双引号），因此写成"""aaa/aaa"""。

例如，使用上面的规则举例，密码为 SYSDBA/,@.:，端口号为 5236 的本地服务器，USERID 可写成如下所示：

```
./dexp USERID=SYSDBA/"""SYSDBA/,@.:"""@LOCALHOST:PORT
```

FILE

FILE 用于明确指定导出的文件（可以包含路径），可以包含多个文件，用逗号分隔。可选参数，如果不选用 FILE 参数（来明确指定导出文件名称），那么默认导出文件名称为 dexp.dmp。

语法如下：

```
FILE=<文件>
```

例如，设置 FILE=db_str.dmp，导出文件名为 db_str.dmp。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log FULL=Y DIRECTORY=/mnt/data/dexp
```

例如，FILE 指定的文件也可以包含路径，设置 FILE=/mnt/db_str.dmp。

```
./dexp USERID=SYSDBA/SYSDBA FILE=/mnt/db_str.dmp LOG=db_str.log FULL=Y
```

文件也可以在生成过程中自动扩展成多个。首先，文件名需要包含通配符%U，用于作为自动扩充文件的文件名模板。%U 表示为 2 个字符宽度的数字，由系统自动生成，起始为 01。其次，使用 FILESIZE 参数来指定文件的大小。

例如，设置 FILE=db_str%U.dmp，导出文件名为 db_str%U.dmp。同时使用%U 对

文件进行扩展，使用 FIZESIZE=128m 指定文件大小。

```
./dexp    USERID=SYSDBA/SYSDBA    FILE=db_str%U.dmp    LOG=db_str.log    FULL=Y
DIRECTORY=/mnt/data/dexpFILESIZE=128m
```

LOG

LOG 用于明确指定导出的日志文件名称（可以包含路径），可以包含多个文件，用逗号分隔。可选参数。如果不选用 LOG 参数（来明确指定日志文件名称），那么默认导出日志文件名称为 dexp.log。

语法如下：

```
LOG=<文件名>
```

例如，指定日志文件名为 db_str.log。

```
./dexp    USERID=SYSDBA/SYSDBA    FILE=db_str.dmp    LOG=db_str.log    FULL=Y
DIRECTORY=/mnt/data/dexp
```

LOG内容具体介绍如下：

1.如果导出级别为TABLES，那么日志信息包含一张或多张表的索引、权限、数据行数。

2.如果导出级别为SCHEMAS，那么日志信息只包含一个或多个模式中的对象统计信息，每个模式下的内容具体有：

- 导出SEQUENCE、VIEW、COMMENT、PROCEDURE、SYNONYM、DBLINK、TRIGGER、PACKAGE、OBJECT、CLASS_BODY和DOMAIN等对象的个数和名称，以及对象的权限。
- 依次导出表的索引、权限、数据行数。

3. 如果导出级别为OWNER，那么日志内容包含一个或多个用户所拥有的所有模式中的对象统计信息，每个模式下的内容和导出级别为SCHEMAS时一样。

4. 如果导出级别为FULL，那么日志内容除了包含所有模式中的对象信息，还包括系统包和系统视图。

达梦数据库信息大赛资料

DIRECTORY

DIRECTORY 用于指定导出文件及日志文件生成的路径。可选参数，缺省为导出到 dexp 所在路径。

语法如下：

```
DIRECTORY=<path>
```

<path>：导出文件和日志文件生成的路径。

如果 FILE 和 LOG 参数指定的文件包含生成路径，则 FILE 和 LOG 参数中指定的路径将替代 DIRECTORY 所指定的路径；如果 FILE 和 LOG 参数指定的文件未包含路径信息，则文件将被生成到 DIRECTORY 指定的目录下；如果都没有指定路径，程序将根据当前的运行环境来设置相应的导出路径，一般为当前路径。

例如，没有指定路径，使用当前路径。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log FULL=Y
```

例如，指定路径/mnt/data/dexp。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log FULL=Y DIRECTORY=/mnt/data/dexp
```

说明：**FILE 和 LOG 中指定的路径，优先级高于 DIRECTORY 中指定的路径。**
当 FILE 和 LOG 不指定路径时，才使用 DIRECTORY 指定的路径。
如果都没有指定路径，则使用系统当前路径。

导出方式

针对数据库对象，有 FULL、OWNER、SCHEMAS、TABLES 四种导出方式可供选择。一次导出只能指定一种方式。可选参数，缺省为 SCHEMAS。这里仅介绍 FULL 方式全库导出。

FULL

FULL 方式导出数据库的所有对象。

语法如下：

```
FULL=Y
```

例如，设置 FULL=Y，导出数据库的所有对象，导出数据库文件和日志文件放在路径/mnt/data/dexp 下。

```
./dexp      USERID=SYSDBA/SYSDBA      FILE=db_str.dmp      LOG=db_str.log
FULL=YDIRECTORY=/mnt/data/dexp
```

6 dimp 逻辑导入

dimp 逻辑导入工具利用 dexp 工具生成的备份文件对本地或远程的数据库进行联机逻辑还原。dimp 导入是 dexp 导出的相反过程。还原的方式可以灵活选择，如是否忽略对象存在而导致的创建错误、是否导入约束、是否导入索引、导入时是否需要编译、是否生成日志等。

6.1 使用 dimp 工具

dimp 工具需要从命令行启动。在 cmd 命令行工具中找到 dimp 所在安装目录/dmdbms/bin，输入 dimp 和参数后回车。参数在下一节详细介绍。

语法如下：

```
dimp  PARAMETER=value { PARAMETER=value }
```

PARAMETER：dimp 参数。多个参数之间排列顺序无影响，参数之间使用空格间隔。

value：参数取值。

例如，将逻辑备份采用 FULL 方式完全导入到用户名和密码为 SYSDBA，IP 地址为192.168.0.248，端口号为 8888 的数据库。导入文件名为 db_str.dmp，导入的日志文件名为 db_str.log，路径为/mnt/data/dexp。

```
./dimp       USERID=SYSDBA/SYSDBA@192.168.0.248:8888       FILE=db_str.dmp
DIRECTORY=/mnt/data/dexp  LOG=db_str.log FULL=Y
```

6.2 dimp 参数一览表

本节提供 dimp 的参数一览表，供用户快速参考。

表 3.1 dimp 参数一览表

参数	含义	备注
USERID	用户名/口令@主库名:端口号#证书路径例如：SYSDBA/SYSDBA*MPP_TYPE @server:5236#ssl_path@ssl_pwd	必选。其中主库名、端口号和证书路径为可选项
FILE	输入文件，即 dexp 导出的文件	必选
DIRECTORY	导入文件所在目录	可选
FULL	导入整个数据库 (N)	可选，四者中选其一。缺省为 SCHEMAS
OWNER	导入指定的用户名下的模式	
SCHEMAS	导入的模式列表	
TABLES	表名列表，指定导入的 tables 名称。不支持对外部表进行导入	
PARALLEL	用于指定导入的过程中所使用的线程数目	可选
TABLE_PARALLEL	用于指定导入的过程中每个表所使用的子线程数目	可选。在 FAST_LOAD 为 Y 时有效
IGNORE	忽略创建错误 (N)。如果表已经存在则向表中插入数据，否则报错表已经存在。	可选
TABLE_EXISTS_ACTION	需要的导入表在目标库中存在时采取的操作 [SKIP| APPEND | TRUNCATE | REPLACE]	可选
FAST_LOAD	是否使用 dmfldr 进行数据导入(N)	可选
FLDR_ORDER	使用 dmfldr 是否需要严格按顺序来导数据(Y)	可选
COMMIT_ROWS	批量提交的行数(5000)	可选
EXCLUDE	忽略指定的对象 (CONSTRAINTS,INDEXES,ROWS,TRIGGERS,GRANTS)。格式	可选。例如 EXCLUDE=(CONSTRAINT)

达梦数据库信创大赛资料

	EXCLUDE=(CONSTRAINTS,INDEXES,ROWS,TRIGGERS,GRANTS)	
GRANTS	导入权限 (Y)	可选
CONSTRAINTS	导入约束 (Y)	可选
INDEXES	导入索引 (Y)	可选
TRIGGERS	导入触发器（Y）	可选
ROWS	导入数据行 (Y)	可选
LOG	日志文件	可选
NOLOGFILE	不使用日志文件(N)	可选
NOLOG	屏幕上不显示日志信息(N)	可选
DUMMY	交互信息处理：打印(P)。取值 Y/N。 Y：打印所有交互信息。N：不打印交互信息。	可选
LOG_WRITE	日志信息实时写入文件(N)	可选
PARFILE	参数文件名，如果 dimp 的参数很多，可以存成参数文件	可选
FEEDBACK	显示每 x 行 (0) 的进度	可选
COMPILE	编译过程, 程序包和函数 (Y)	可选
INDEXFILE	将表的索引/约束信息写入指定的文件	可选
INDEXFIRST	导入时先建索引(N)	可选
REMAP_SCHEMA	SOURCE_SCHEMA：TARGET_SCHEMA 将 SOURCE_SCHEMA 中的数据导入到 TARGET_SCHEMA 中	可选
ENCRYPT_PASSWORD	数据的加密密钥	可选。和 dexp 中的 ENCRYPT_PASSWORD 设置的密钥一样
ENCRYPT_NAME	数据的加密算法的名称	可选。和 dexp 中的 ENCRYPT_NAME 设置的加密算法一样

达梦数据库信创大赛资料

SHOW/ DESCRIBE	只列出文件内容 (N)	可选
LOCAL	MPP 环境下，是否使用 LOCAL 方式登录(N)	可选
TASK_THREAD_NUMBER	设置 dmfldr 处理用户数据的线程数目	可选
BUFFER_NODE_SIZE	设置 dmfldr 读入文件缓冲区大小	可选
TASK_SEND_NODE_NUMBER	用于设置 dmfldr 发送节点个数[16,65535]	可选
LOB_NOT_FAST_LOAD	如果一个表含有大字段，那么不使用 dmfldr，因为 dmfldr 是一行一行提交的	可选
PRIMARY_CONFLICT	主键冲突的处理方式[IGNORE|OVERWRITE],默认报错	可选
TABLE_FIRST	是否强制先导入表 (默认 N),Y 表示先导入表,N 正常导入	可选
HELP	显示帮助信息	可选

6.3 dimp 参数详解

6.3.1 特殊参数

特殊参数中介绍了必选参数的和最常用的可选参数。

USERID

USERID 用于连接服务器，必选参数。用法请参考 dexp 中 USERID。

FILE

FILE 用于指定明确导入 dexp 导出的文件（可以包含路径）。为必选参数。

语法如下：

```
FILE=<文件>
```

例如，之前导出的文件路径为：FILE=/mnt/data/dexp/db_str.dmp，那么导入文件可以写成如下所示：

```
./dimp  USERID=SYSDBA/SYSDBA  FILE=/mnt/data/dexp/db_str.dmp  LOG=db_str.log
DIRECTORY=/mnt/data/dimp
```


达梦数据库信创大赛资料

若 FILE 没有指定文件路径，则可在 DIRECTORY 中指定。

LOG

LOG 用于明确指定导入的日志文件名称（可以包含路径），可以包含多个文件，用逗号分隔。可选参数。如果不选用 LOG 参数（来明确指定日志文件名称），那么默认导出日志文件名称为 dimp.log。

语法如下：

```
LOG=<文件名>
```

例如，指定日志文件名为 db_str.log。

```
./dimp  USERID=SYSDBA/SYSDBA  FILE=/mnt/data/dexp/db_str.dmp  LOG=db_str.log
FULL=Y DIRECTORY=/mnt/data/dimp
```

LOG文件记载了导出对象的统计信息。供用户查看导出内容。详细的日志内容和dexp LOG中一样，请参考dexp的LOG。

DIRECTORY

DIRECTORY 用于指定导入文件及日志文件生成的路径。可选参数，缺省为 dimp 所在路径。

语法如下：

```
DIRECTORY=<path>
```

<path>：导入文件路径和日志文件生成的路径。

如果 FILE 和 LOG 参数指定的文件包含生成路径，则 FILE 和 LOG 参数中指定的路径将替代 DIRECTORY 所指定的路径；如果 FILE 和 LOG 参数指定的文件未包含路径信息，则导入文件路径和日志生成的文件路径都是 DIRECTORY 指定的路径；如果都没有指定路径，程序将根据当前的运行环境来设置相应的导出路径，一般为当前路径。

例如，没有指定路径，使用当前路径。

```
./dimp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log FULL=Y
```

例如，指定 FILE 使用/mnt/data/dexp 路径，LOG 使用/mnt/data/dimp 路径。

```
./dimp  USERID=SYSDBA/SYSDBA  FILE=/mnt/data/dexp  db_str.dmp  LOG=db_str.log
FULL=Y DIRECTORY=/mnt/data/dimp
```

说明：

FILE 和 LOG 中指定的路径，优先级高于 DIRECTORY 中指定的路径。

当 FILE 和 LOG 不指定路径时，才使用 DIRECTORY 指定的路径。

如果都没有指定路径，则使用系统当前路径。

导入方式

针对数据库对象，有 FULL、OWNER、SCHEMAS、TABLES 四种导入方式可供选择。一次导出只能指定一种方式。可选参数，缺省为 SCHEMAS。这里仅介绍 FULL 方式全库导出。

小窍门：

导入方式可以和导出方式不一致。

例如：当导出方式不为 FULL 时，导入时也可以指定为 FULL 方式，此时的 FULL 会将导出文件包含的全部内容导入到指定的实例中。

FULL

FULL 方式导入整个数据库。

语法如下：

```
FULL=Y
```

例如，设置 FULL=Y，导入整个数据库，导入的数据库文件在/mnt/data/dexp，即将生成的日志文件放在/mnt/data/dimp。

```
./dimp  USERID=SYSDBA/SYSDBA  FILE=/mnt/data/dexp/db_str.dmp  LOG=db_str.log
FULL=Y DIRECTORY=/mnt/data/dimp
```

7 dexp 和 dimp 应用实战

7.1 一个完整示例

下面列出一个完整的导出/导入示例以供参考。

1. 环境准备

导出库：环境为 linux，服务器为 192.168.0.248，用户名和密码均为 SYSDBA。导出的是 DM 数据库系统安装时自带的名为 BOOKSHOP 的示例库，端口号 5236。

达梦数据库信创大赛资料

导入库：环境为 linux，服务器为 192.168.0.248，用户名和密码均为 SYSDBA。准备一个空数据库作为导入库，端口号为 8888。

2. dexp 导出

导出数据库的所有对象（FULL=Y），导出文件为 dexp01.dmp，导出日志为 dexp01.log，导出文件和日志文件都存放在/emc_2/data/dexp 目录中。

```
./dexp SYSDBA/SYSDBA@192.168.0.248:5236 FILE=dexp01.dmp LOG=dexp01.log DIRECTORY=/emc_2/data/dexp FULL=Y
```

3. dimp 导入

导入 SYSDBA、OTHER、PERSON 模式中的数据（SCHEMAS=SYSDBA,OTHER,PERSON），导入文件就是上一步导出的文件 dexp01.dmp，导入日志 dimp02.log 放入/emc_2/data/dimp 目录中。

```
./dimp SYSDBA/SYSDBA@192.168.0.248:8888 FILE=/emc_2/data/dexp/dexp01.dmp LOG=dimp01.log DIRECTORY=/emc_2/data/dimp FULL=y
```

4. 查看 LOG 日志

如果想了解详细的导出、导入内容统计信息，请查看 LOG 日志 dexp01.log 和 dimp01.log。

达梦数据库信创人才资料